Proceedings

TWENTY-SIXTH ANNUAL ALLERTON CONFERENCE ON COMMUNICATION, CONTROL, AND COMPUTING

September 28-30, 1988

Allerton House, Monticello, Illinois
Sponsored by the
Coordinated Science Laboratory and the
Department of Electrical and Computer Engineering of the
University of Illinois at Urbana-Champaign

E[illegible]NEERING LIBRARY
U[illegible]SITY OF ILLINOIS
U[illegible]BANA, ILLINOIS

FEB 0 7 1989

NOTICE: **Return or renew all Library Materials! The *Minimum* Fee for each Lost Book is $50.00.**

The person charging this material is responsible for its return to the library from which it was withdrawn on or before the **Latest Date** stamped below.

Theft, mutilation, and underlining of books are reasons for disciplinary action and may result in dismissal from the University.
To renew call Telephone Center, 333-8400

UNIVERSITY OF ILLINOIS LIBRARY AT URBANA-CHAMPAIGN

ENGINEERING

MAR 2 0 1989
NOV 0 2 1995
REC'D FEB 27 '89
NOV [illegible] REC'D
APR 13 1989
MAR 3 0 REC'D
INTERLIBRARY LOAN
DUE: 5/25/89
6-27-89
JUN 9 REC'D
JUL 7 REC'D
PHOTO REPRODUCTION
AUG 3 REC'D
INTERLIBRARY LOAN
OCT 2 0

L161—O-1096

PROCEEDINGS

TWENTY-SIXTH ANNUAL ALLERTON CONFERENCE ON COMMUNICATION, CONTROL, AND COMPUTING

VOLUME I

Mark W. Spong
Bruce Hajek
Conference Co-Chairmen

Conference held
September 28-30, 1988
Allerton House
Monticello, Illinois

Sponsored by
The Coordinated Science Laboratory
and
The Department of Electrical and Computer Engineering
of the
UNIVERSITY OF ILLINOIS
at
Urbana-Champaign

TABLE OF CONTENTS

FOREWORD i

I-A: ESTIMATION AND DETECTION

Chairman: J. Sacks
University of Illinois at Urbana-Champaign

AN OPTIMUM SEQUENCE ESTIMATION APPROACH IN THE ABSENCE OF SYNCHRONIZATION 1
C. N. Georghiades

COMPUTATION OF THE MINIMUM-ASN BOUNDARIES FOR THE MODIFIED SPRT: PERFORMANCE ANALYSIS AND COMPARISONS 11
O. Snorrason and F. D. Garber

A PARAMETRIC FORM FOR SIEVE-CONSTRAINED MAXIMUM-LIKELIHOOD DENSITY ESTIMATION FROM NOISY DATA 13
T. J. Schulz and D. L. Snyder

ASYMPTOTICALLY EFFICIENT ESTIMATION OF PRIOR PROBABILITIES IN MULTICLASS FINITE MIXTURES 15
G. R. Dattatreya and L. N. Kanal

ROBUST LOCALLY-OPTIMAL DETECTION OF DEPENDENT SIGNALS IN NON-GAUSSIAN NOISE 25
P. Willett and P. Srivastava

CRAMER-RAO BOUNDS FOR CONSTRAINED SPECTRUM ESTIMATION WITH APPLICATION TO A PROBLEM IN RADAR IMAGING 27
J. A. O'Sullivan, P. Moulin and D. L. Snyder

ROBUSTNESS IN TERMS OF A J-DIVERGENCE SIGNAL-TO-NOISE RATIO 35
D. Warren

RESISTANCE AND ROBUSTNESS FOR FINITE MEMORY DETECTORS 37
K. S. Vastola

DECISION FUSION IN DISTRIBUTED DETECTION WITH UNCERTAINTIES 39
E. Drakopoulos and C. C. Lee

I-B: COMPUTATION GEOMETRY

Chairman: Franco Preparata
University of Illinois at Urbana-Champaign

THE NUMBER OF MAXIMAL EDGE SEQUENCES AS A CONVEX POLYTOPE 49
C. Schevon and J. O'Rouke

TWO TECHNIQUES FOR CREATING EXTRA COMPONENTS TO OBTAIN LOWER BOUNDS 58
P. Ramanan

A SEARCH PROBLEM IN PLANE GEOMETRY 63
K. Sugihara, I. Suzuki and M. Yamashita

GEOMETRIC SELECTION BY PARAMETRIC SEARCH 73
J. S. Salowe

AN OPTIMAL ALGORITHM FOR CONSTRAINED DELAUNAY TRIANGULATION 83
D.-M. Jung

LOWER BOUNDS FOR SOME CONSTRAINED VORONOI DIAGRAMS 85
C. A. Wang

A UNIFIED APPROACH TO FINDING ENCLOSING AND ENCLOSED TRIANGLES 87
D. M. Mount and S. Chandran

COVERING ORTHOGONAL POLYGONS WITH SQUARES 97
L. J. Aupperle, H. E. Conn, J. M. Keil and J. O'Rourke

AN EFFICIENT ON-LINE LINEAR CHEBYSHEV APPROXIMATION ALGORITHM FOR A FINITE PLANAR POINT SET 107
B. K. Bhattacharya

I-C: MULTIPROCESSOR ARCHITECTURES AND ALGORITHMS

Chairman: Wen-Mei Hwu
University of Illinois at Urbana-Champaign

DESIGN AND ANALYSIS OF A HIERARCHICAL SNOOPING CACHE COHERENCE SYSTEM 109
M. Azimi and C. Erickson

A PERFORMANCE STUDY FOR PARALLEL COMPUTATION ON LOOSELY-COUPLED MULTIPROCESSOR SYSTEMS 119
L. Liu, J.-K. Peir, A. Bross and D. Shen

A MAPPING ALGORITHM FOR MULTIPROCESSOR ARCHITECTURES 129
S. Yalamanchili and D. T. Lee

SYSTEMS WITH NEARLY ZERO DISTRIBUTED SIMULATION OVERHEAD 139
D. Kumar

DISTRIBUTED COMPUTATION ON CONCURRENT PROCESSORS TIME SHARED MULTICOMPUTING 149
V. K. Madisetti and D. G. Messerschmitt

I-D: LINEAR SYSTEMS

Chairman: Juraj V. Mendaic
University of Illinois at Urbana-Champaign

NEW RESULTS ON EIGENVALUE CONCEPTS FOR LINEAR TIME VARYING DYNAMICAL SYSTEMS 159
J. Zhu and C. D. Johnson

IMPLUSE RESPONSES AND KERNEL REPRESENTATIONS 169
D. Ball and I. W. Sandberg

CLASSICAL AND EXTENDED TRANSMISSION ZEROS: MODEL MATCHING RELATIONSHIPS 177
M. K. Sain, B. F. Wyman and J. L. Peczkowski

AN EXTENSION OF KHARITONOV'S THEOREM FOR A CLASS OF MULTILINEAR COEFFICIENT PERTURBATIONS 187
B. R. Barmish and Z. Shi

INVARIANT SUBZEROS ARE NOT INVARIANT 189
C. B. Schrader and M. K. Sain

ON STABLE SOLUTIONS OF THE ONE AND TWO SIDED MODEL MATCHING PROBLEMS 191
Z. Gao and P. J. Antsaklis

SYSTEMS IDENTIFICATION AND MODEL REDUCTION OF LINEAR CONTINUOUS SYSTEMS FROM IMPULSE RESPONSE DATA 199
P. Agathoklis and V. Sreeram

NEAR-OPTIMAL MODEL REDUCTION IN BALANCED AND MODAL COORDINATES 209
W. Gawronski and J.-N. Juang

I-E: DIGITAL SIGNAL PROCESSING

Chairman: K. S. Arun
University of Illinois at Urbana-Champaign

A PARAMETRIC METHOD FOR DETERMINING THE NUMBER OF SIGNALS 219
Q. Wu and D. R. Fuhrmann

AN EXTENDED OBE ALGORITHM FOR ARMA PARAMETER ESTIMATION 229
A. K. Rao, Y.-F. Huang and S. Dasgupta

NONLINEAR ITERATIVE TECHNIQUES FOR IMAGE RESTORATION FROM INCOMPLETE SPECTRAL DATA 239
K. D. Sanders and A. C. Bovik

EIGENANALYSIS BASED BISPECTRUM ESTIMATION 249
D. L. Sherman and M. D. Zoltowski

OPTIMAL DISCRIMINATION OF MULTIDIMENSIONAL AUTOREGRESSIVE PROCESSES 251
M. M. Milosavljevic

II-A: PARALLEL ALGORITHMS

Chairman: Vijaya Ramachandran
University of Illinois at Urbana-Champaign

EFFICIENT PARALLEL ALGORITHMS FOR STRING EDITING AND RELATED PROBLEMS 253
A. Apostolico, M. J. Atallah, L. L. Larmore and H. S. McFaddin

FAST PARALLEL ALGORITHMS FOR BIPARTITE PERMUTATION GRAPHS 264
L. Chen and Y. Yesha

PT-OPTIMAL ALGORITHMS FOR INTERVAL GRAPHS 274
A. Moitra and R. Johnson

FAST PARALLEL ALGORITHMS FOR TREE DECOMPOSING AND PARSING PARTIAL k-TREES 283
N. Chandrasekharan and S. T. Hedetniemi

AN OPTIMAL PARALLEL ALGORITHM FOR PREEMPTIVE JOB SCHEDULING THAT MINIMIZES MAXIMUM LATENESS 293
S. H. Rodger

FINDING A MINIMAL COVER FOR BINARY IMAGES: AN OPTIMAL PARALLEL ALGORITHM 303
D. Moitra

II-B: NONLINEAR SYSTEMS

Chairman: Ren Jeng Su
University of Colorado at Boulder

STABILITY ANALYSIS OF ALMOST SINUSOIDAL PERIODIC OSCILLATIONS IN NONLINEAR CONTROL SYSTEMS SUBJECTED TO NONCONSTANT PERIODIC INPUT 313
G. S. Krenz and R. K. Miller

PULSE-WIDTH-MODULATION CONTROLLER DESIGN FOR NONLINEAR DELAY DIFFERENTIAL SYSTEMS 321
H. Sira-Ramirez

DESIGN OF NONCHAOTIC FEEDBACK SYSTEMS 331
V. Raman and R.-W. Liu

LIAPUNOV FUNCTIONS FOR CRITICAL NONLINEAR SYSTEMS 338
J.-H. Fu and E. H. Abed

ON STABILIZATION WITH A PRESCRIBED REGION OF ASYMPTOTIC STABILITY 348
L. Saydy, E. H. Abed and A. L. Tits

APPLICATION OF HAMILTONIAN SYSTEM THEORY TO THREE-MACHINE POWER SYSTEM TRANSIENT ANALYSIS 356
T. Guo, N. Narasimhamurthi and N. H. McClamroch

IDENTIFICATION OF A CLASS OF PARALLEL SYSTEMS 366
M. Pawlak

II-C: FAULT TOLERANCE AND DIAGNOSIS

Chairman: Kent Fuchs
University of Illinois at Urbana-Champaign

USING REPLICATION TO IMPLEMENT RELIABLE STORAGE 368
B. Bhargava and S. Browne

CORRESPONDENT COMPUTING BASED SOFTWARE FAULT TOLERANCE 378
P.-N. Lee, A. Tamboli and J. Blankenship

IMPROVED UPPER BOUNDS FOR ALGORITHM-BASED FAULT TOLERANCE 388
D. J. Rosenkrantz and S. S. Ravi

ADAPTIVE DIAGNOSIS FOR PROBABILISTICALLY DIAGNOSABLE SYSTEMS 398
F.-H. W. Shih and K. Nakajima

PROPERTIES OF c-CORRECTABILITY IN SELF-DIAGNOSING SYSTEMS 408
J. T. Butler and Y. Rosen

II-D: ADAPTIVE AND ARRAY SIGNAL PROCESSING

Chairman: W. K. Jenkins
University of Illinois at Urbana-Champaign

AVERAGING THEORY FOR SIGN-SIGN LMS 417
B. D. O. Anderson, I. M. Y. Mareels, W. A. Sethares and C. R. Johnson

ON MEAN SQUARE CONVERGENCE OF ADAPTIVE BEAM-FORMING ALGORITHMS 427
G. Yin

AN EFFICIENT ALGORITHM FOR TRACKING MULTIPLE MOVING TARGETS 437
C. R. Sastry, E. W. Kamen and M. Simaan

INSUFFICIENT ORDER ADAPTIVE IIR FILTERS WITH WHITE INPUTS 447
M. Nayeri

THE PERFORMANCE CHARACTERISTICS OF THE ORDER RECURSIVE ADAPTIVE LATTICE EQUALIZER 457
B. M. Sallam and S. B. Kesler

ADAPTIVE AVERAGE FILTERS FOR IMAGE PROCESSING 459
B. Chang and M. Kanefsky

II-E: INFORMATION THEORY AND CODING

Chairman: D. L. Snyder
Washington University

ASYMPTOTIC BOUNDS FOR FOCUSED ERROR CONTROL CODES 461
T. Fuja and C. Heegard

THE ENUMERATION OF (D,K) SEQUENCES 471
K. Forsberg and I. Blake

FOURIER TRANSFORM CODING FOR PEAK LIMITED CHANNELS 473
E. Feig

IMPROVING GENERALIZED MINIMUM DISTANCE DECODING BY COMPARISON WITH ERRORS-ONLY DECODING 480
D. Taipale

A UNIVERSAL OUTER CODE DECODING TECHNIQUE 482
J. J. Metzner

AN IMPROVED DECODING ALGORITHM FOR CERTAIN ELLIPTIC CODES 492
C.-M. Liu and P. V. Kumar

OPTICAL ORTHOGONAL CODES - NEW BOUNDS AND AN OPTIMAL CONSTRUCTION 494
H. Chung and P. V. Kumar

MAXIMUM MUTUAL INFORMATION DECODING FOR ARBITRARILY VARYING MULTIPLE-ACCESS CHANNELS 499
J. A. Gubner

III-A: VARIALBE STRUCTURE CONTROL AND ROBOTICS

Chairman: Ray DeCarlo
Purdue University

BINARY CONTROL SYSTEMS DESIGN 501
R. T. Yanushevsky

ANALYSIS AND SYNTHESIS OF A CLASS OF NEURAL NETWORKS: VARIABLE STRUCTURE SYSTEMS WITH INFINITE GAINS 508
J.-H. Li, A. M. Michel and W. Porod

ON THE ESTIMATION OF SLIDING DOMAINS AND STABILITY REGIONS OF VARIABLE STRUCTURE CONTROL SYSTEMS 518
S. M. Madani-Esfahani, S. Hui and S. H. Zak

THE ROLE OF SLIDING MODE IN CONTACT FORCE CONTROL OF ROBOT MANIPULATORS 528
K. D. Young

ON CONTINUOUS APPROXIMATION TO VARIABLE STRUCTURE CONTROL 535
F. Esfandiari and H. K. Khalil

STABILITY OF A CLASS OF LINEAR MECHANICAL SYSTEMS WITH LINEAR HOLONOMIC CONSTRAINTS 537
D. Wang and N. H. McClamroch

TIP POSITION ADAPTIVE CONTROL OF A FLEXIBLE MANIPULATOR 547
T.-C. Yang, J. C. S. Yang and P. Kudva

III-B: STOCHASTIC METHODS

Chairman: W. A. Massey
AT&T Bell Laboratories

COMPENSATORS AND DERIVATIVE ESTIMATION FOR QUEUEING SYSTEMS 549
M. A. Zazanis

MONTE CARLO SIMULATION AND LARGE DEVIATIONS THEORY: THE ASYMPTOTIC OPTIMALITY OF THE EXPONENTIAL SHIFT 556
J. S. Sadowsky, J. A. Bucklew and P. Ney

CUSTOMER ROUTING TO DIFFERENT SERVERS WITH COMPLETE INFORMATION 566
Z. Rosberg and P. Kermani

AVERAGE COST SEMI-MARKOV DECISION PROCESSES AND THE CONTROL OF QUEUEING SYSTEMS 569
L. I. Sennott

QUICKER CONVERGENCE FOR OPTIMAL STOPPING PROBLEMS THROUGH PROBABILISTIC INTERPRETATIONS OF CLASSICAL NUMERICAL METHODS 571
A. G. Greenberg and R. J. Vanderbei

ENTROPY RATE OF STATIONARY RANDOM FIELDS ON BINARY TREE 580
T. Berger and Z. Ye

MATRIX QUADRATIC EQUATIONS AND QUASI-BIRTH-AND-DEATH MODELS OF MULTIPLE ACCESS NETWORKS 589
G. R. Murthy and E. J. Coyle

ON FUSION IN ESTIMATION THEORY 599
E. B. Hall, A. E. Wessel and G. L. Wise

III-C: IMAGE PROCESSING AND ANALYSIS

Chairman: Yoram Bresler
University of Illiniois at Urbana-Champaign

IMPROVING COVER TYPE IDENTIFICATION IN SPECKLED SAR IMAGES BY PREFILTERING AND SEQUENTIAL CLASSIFICATION 609
Q. Lin and J. Allebach

ESTIMATING CONVEX SHAPES FROM SUPPORT LINE MEASUREMENTS USING PRIOR GEOEMTRIC INFORMATION 619
J. L. Prince and A. S. Willsky

A TWO-DIMENSIONAL AUTOREGRESSIVE MODELING TECHNIQUE FOR ANALYSIS AND CLASSIFICATION OF PLANAR SHAPES 629
M. Das, M. J. Paulik and N. K. Loh

UNCERTAINTY AND SIMILARITY MEASURES IN PATTERN RECOGNITION 639
J. Dj. Golic

A MODEL AND SYNTHESIS OF TEXTURE ON A 3-D SURFACE 647
Y. Choe and R. L. Kashyap

EDGE DETECTION BY SIMULATED ANNEALING 657
H. L. Tan and E. J. Delp

STATISTICAL APPROACH TO NONDIFFRACTION COMPUTED TOMOGRAPHY IMAGING AND ITS APPLICATION IN IMAGE ANALYSIS 659
T. Lei and W. Sewchand

SIMULTANEOUS IDENTIFICATION AND RESTORATION OF IMAGES USING MAXIMUM LIKELIHOOD ESTIMATION AND THE EM ALGORITHM 661
K. T. Lay and A. K. Katsaggelos

III-D: DISTRIBUTED COMPUTATION AND COMPLEXITY THEORY

Chairman: M. C. Loui
University of Illinois at Urbana-Champaign

EFFICIENT AGREEMENT USING FAULT DIAGNOSIS 663
B. A. Coan

ON EFFICIENTLY COMPUTING FUNCTIONS OF DISTRIBUTED INFORMATION 673
T. V. Lakshman and V. K. Wei

ON FINDING STRONGLY CONNECTED COMPONENTS DISTRIBUTIVELY 683
K. Brahmadathan and S. Venkatesan

IMPROVING PETERSON'S CONSTRUCTION OF 1-WRITER n-READER MULTIVALUED ATOMIC REGISTER 693
K. Vidyasankar

AN ASYNCHRONOUS ALL PAIRS SHORTEST PATH ALGORITHM FOR SHARED MEMORY MULTIPROCESSORS 701
A. Uresin and M. Dubois

TEST CASE CONSTRUCTION FOR NP-HARD PROBLEMS 711
L. A. Sanchis

RELATIVE COMPLEXITY OF EVALUATING AND CONSTRUCTING THE OPTIMUM OF MAXIMIZATION PROBLEMS 721
P. Crescenzi and R. Silvestri

III-E: TESTING AND CONCURRENT ERROR DETECTION

Chairman: J. Patel
University of Illinois at Urbana-Champaign

TESTING OF CMOS CIRCUITS AT FUNCTIONAL LEVEL 731
L. Zhang, L. Tong and R.-W. Liu

ON THE COMPLEXITY OF COMPUTING TESTS FOR CMOS GATES 740
S. Chakravarty

A HIERARCHICAL SEQUENTIAL TEST GENERATOR 750
W.-T. Cheng and J.-Y. Jou

ON THE DESIGN OF VLSI CMOS PLAs FOR BIST IMPLEMENTATION 758
R. Dandapani, R. K. Gulati, D. K. Goel and K. R. Callahan

SFS/SSC DOMINO-CMOS IMPLEMENTATIONS OF TSC CIRCUITS 768
N. K. Jha

FAULT COLLAPSING FOR CONCURRENT ERROR DETECTION ANALYSIS 778
L. L. Kinney and T. Ramakrishnan

IV-A: COMBINATORIAL ALGORITHMS

Chairman: C. L. Liu
University of Illinois at Urbana-Champaign

SEPARATORS, GRAPH HOMOMORPHISMS AND CHROMATIC POLYNOMIALS 788
H. B. Hunt, III, S. S. Ravi and R. E. Stearns

ALGORITHMS FOR BISECTING PLANAR GRAPHS 798
T. N. Bui and A. Peck

A BETTER CLIQUE-PARTITIONING ALGORITHM 808
J. Bhasker and T. Samad

ADAPTIVE AND OPTIMAL RANDOMIZATION OF ALGORITHMS IN P: A CASE STUDY 810
W. H. Janko, A. Geyer-Schulz and A. Taudes

THE SET UNION PROBLEM WITH DYNAMIC WEIGHTED BACKTRACKING 819
G. Gambosi, G. F. Italiano and M. Talamo

FINDING NORMAL FORMS, CLIQUES, HAMILTONIAN PATHS AND HAMILTONIAN CIRCUITS IN A SET OF PROPER CIRCULAR ARCS IN OPTIMUM TIME AND SPACE 832
G. K. Manacher and T. A. Mankus

LINEAR TIME ALGORITHMS ON CIRCULAR-ARC GRAPHS 842
W.-L. Hsu and K. Tsai

IV-B: CONTROL SYSTEMS DESIGN

Chairman: S. H. Zak
Purdue University

NEW ESTIMATES OF ASYMPTOTIC TRAJECTORY BOUNDS FOR DIGITAL FILTERS: IMPROVED RESULTS 852
J. A. Farrell and A. N. Michel

A GENERALIZED LEAD-LAG COMPENSATION TECHNIQUE 862
N. M. Arbouz and R. Su

A POLE-PLACEMENT PROCEDURE VIA H^{∞}-OPTIMIZATION 872
A. E. Frazho and D. K. Le

ROBUSTNESS OF FEEDBACK SYSTEMS WITH SEVERAL SOURCES OF MODELLING ERROR 881
J. S. Freudenberg and J. Chen

REPRESENTATION OF UNCERTAIN ACTUATOR DYNAMICS AND DECENTRALIZED ROBUST CONTROLLER DESIGN 883
A. Iftar and U. Ozguner

TIME DOMAIN CONTROL DESIGN FOR ROBUST STABILITY OF DISCRETE-TIME LINEAR REGULATORS 893
S. R. Kolla and J. B. Farison

SPECTRAL APPROXIMATION OF ALMOST BLOCK-TRIANGULAR OPERATORS AND HANKEL-NORM APPROXIMATION OF TWO FREQUENCY SCALE TRANSFER MATRICES 903
H. Oloomi and M. E. Sawan

BLOCK TRIANGULARIZATION OF DISCRETIZED MULTIPARAMETER SINGULARLY PERTURBED SYSTEMS 905
G. L. Syrmos and E. H. Abed

IV-C: STOCHASTIC AND ADAPTIVE CONTROL

Chairman: Han-Fu Chen
University of Illinois at Urbana-Champaign

AN ADAPTIVE CONTROLLER TO ACHIEVE AN ARBITRARILY GOOD TRANSIENT AND STEADY-STATE RESPONSE 907
D. E. Miller and E. J. Davison

A ROBUST ADAPTIVE CONTROL FOR LINEAR PLANTS IN THE PRESENCE OF UNMODELED DYNAMICS 917
S. Bai and F. M. A. Salam

LIAPUNOV BASED MODEL REFERENCE LEARNING CONTROL 927
M. Phan and R. W. Longman

ON LINEAR SYSTEMS DRIVEN BY SELF-EXCITED POINT PROCESSES 937
M. A. Ingram and A. H. Haddad

ADAPTIVE STABILIZATION OF MULTI-MACHINE POWER SYSTEMS 939
H. Yousef and M. Simaan

THE EQUIVALENCE OF STABILIZABILITY CONDITIONS FOR STOCHASTIC SYSTEMS 948
E. Yaz

DRIFT TRANSFORMATIONS USING FEEDBACK CONTROLS AND APPLICATIONS TO SMALL-NOISE PROBLEMS 953
E. Kappos

IV-D: COMMUNICATION NETWORKS

Chairman: L. I. Sennott
Illinois State University

A SPLITTING ALGORITHM FOR COMMUNICATION WITH REAL-TIME DELIVERY CONSTRAINTS 955
D. C. Marinescu

ON THE STABILITY CONDITION OF A PRECEDENCE-BASED QUEUEING DISCIPLINE 965
F. Baccelli and Z. Liu

SIMPLE, EFFICIENT COMPUTATION OF THE GENERALIZED THROUGHPUT OF JACKSON NETWORKS 967
A. G. Greenberg and W. A. Massey

A LOAD-CONTROLLED TOKEN RING PROTOCOL 969
Y. W. Chiang and C. C. Lee

USE OF FLOW CONTROL IN A FINITE USER RANDOM ACCESS SYSTEM................................ 979
A. B. Behroozi-Toosi and R. R. Rao

PARALLEL TIME COMPLEXITY OF ROUTING IN PERMUTATION NETWORKS................................ 981
D. M. Koppelman and A. Y. Oruc

LIMITED WAITING FOR VALUABLE RESOURCES: AN ADAPTIVE CONTROL STRATEGY FOR CIRCUIT-SWITCHED NETWORKS................................ 991
G. Gopal and A. Weinrib

VOICE TRANSMISSION IN PACKET SWITCHING NETWORKS: A MODEL FOR QUEUEING ANALYSIS................................ 1001
F. M. Brochin and J. B. Thomas

IV-E: ALGORITHMS AND ARCHITECTURES FOR SIGNAL AND IMAGE PROCESSING

Chairman: Hua Lee
University of Illinois at Urbana-Champaign

A NEW APPROACH TO SOLVE THE SEQUENCE-LENGTH CONSTRAINT PROBLEM IN CIRCULAR CONVOLUTION USING NUMBER THEORETIC TRANSFORM................................ 1005
H. Lu and S. C. Lee

A FAMILY OF OPTIMAL ARCHITECTURES FOR MULTIDIMENSIONAL TRANSFORMS................................ 1015
C. Chakrabarti and J. JaJa

PARALLEL ARCHITECTURE FOR BOUNDARY FOLLOWING OF REGIONS OF AN IMAGE STORED AS LINEAR QUADTREE................................ 1025
R. B. Panwar and S. K. Nandy

OPTIMAL VISIBILITY ALGORIITHMS FOR BINARY IMAGES ON THE HYPERCUBE................................ 1035
F. Dehne, Q. T. Pham and I. Stojmenovic

A VLSI PARALLEL PROCESSOR FOR BICUBIC SURFACE PATCH IMAGE GENERATION................................ 1037
P. C. Chao and M.-Y. Chern

GRAMMARS AND BAYES' PRIORS FOR HIERARCHICAL IMAGE PROGESSING ON MASSIVELY PARALLEL PROCESSORS................................ 1040
B. Roysam and M. I. Miller

A ROTATION BASED ALGORITHM FOR FAST LEAST SQUARE DECONVOLUTION................................ 1042
Y. H. Hu and P. H. Milenkovic

RESIDUE COMPLEX MULTIPLIERS................................ 1044
A. Skavantzos and T. Stouraitis

EFFICIENT IMPLEMENTATION OF CHIRP Z-TRANSFORM USING CORDIC ALGORITHM................................ 1046
Y. H. Hu and S. Naganathan

PLENARY LECTURE

NEW RESULTS FOR MAGNETIC AND OPTICAL RECORDING CHANNELS 1048
J. K. Wolf
University of California at San Diego

V-A: TRELLIS CODING

Chairman: W. E. Stark
University of Michigan

ON THE PATH WEIGHT ENUMORATORS OF CONVOLUTIONAL CODES 1049
C. C. Chao and R. J. McEliece

AN ARBITRARILY TIGHT UNION BOUND FOR CONVOLUTIONAL CODES THROUGH BIFURCATION 1059
M. V. Burnashev, A. S. Arora and D. L. Cohn

CONSTRUCTION OF GOOD DISTANCE PROFILE TRELLIS CODES FOR SEQUENTIAL DECODING 1061
S. S. Malladi, D. J. Costello, Jr. and H. C. Ferreira

THE APPLICATION OF TRELLIS CODING TECHNIQUES TO DIRECT-SEQUENCE SPREAD-SPECTRUM MULTIPLE ACCESS COMMUNICATIONS 1070
W. E. Stark and B. D. Woerner

UNIFORM DISTANCE AND ERROR PROBABILITY PROPERTIES OF TCM SCHEME 1080
E. Biglieri and P. J. McLane

CONSTRUCTION OF TRELLIS-DECODABLE ERROR-CORRCTING LINE CODES 1082
M. A. Herro, V. Telang and A. R. Calderbank

V-B: PARALLEL COMPUTATION

Chairman: L. B. Pitt
University of Illinois at Urbana-Champaign

DETERMINISTIC SIMULATIONS OF PRAMS ON BOUNDED DEGREE NETWORKS 1084
K. T. Herley and G. Bilardi

MAPPING PYRAMID ALGORITHMS INTO HUYPERCUBES 1094
T.-H. Lai and W. White

OPTIMIZATION OF COMPUTATION TIME FOR SYSTOLIC ARRAYS 1104
Y. Wong and J.-M. Dolosme

TOPOLOGICAL NUMBERING OF FEATURES ON A MESH 1114
M. J. Atallah, S. E. Hambrusch and L. E. TeWinkel

SOLUTION OF LINEAR EQUATIONS ON SHUFFLE-EXCHANGE AND MODIFIED SHUFFLE EXCHANGE NETWORKS 1116
R. B. Panwar and L. M. Patnaik

V-C: VLSI LAYOUT

Chairman: S. S. Ravi
State University of New York at Albany

EMBEDDING RECTANGULAR GRIDS INTO SQUARE GRIDS WITH DILATION TWO 1126
R. Melhem and G.-Y. Hwang

GENERAL CIRCULAR PERMULATION LAYOUT 1136
R. D. Lou and M. Sarrafzadeh

THE ASSIGNMENT HEURISTIC FOR CROSSING REDUCTION IN BIPARTITE GRAPHS 1138
T. Catarci

V-D: DISCRETE EVENT SYSTEMS AND NEURAL NETWORKS

Chairman: D. L. Elliott
National Science Foundation

A STATE-VARIABLE APPROACH TO THE MODELING AND CONTROL OF DISCRETE-EVENT SYSTEMS 1140
Y. Li and W. M. Wonham

MODELING OF INTERACTING CONTINUOUS TIME AND DISCRETE EVENT SYSTEMS: AN EXAMPLE 1150
P. Peleties and R. DeCarlo

BRANCHING TIME TEMPORAL LOGIC FOR DISCRETE EVENT SYSTEM ANALYSIS 1160
K. M. Passino and P. J. Antsaklis

A DECOMPOSITION APPROACH FOR SENSITIVITY ANALYSIS OF DISCRETE-EVENT SYSTEMS 1170
S. G. Strickland and C. G. Cassandras

NEURAL COMPUTING FOR INFORMATION EXTRACTION IN CONTROL SYSTEMS 1172
K. M. Passino, M. A. Sartori and P. J. Antsaklis

MORE ON PARETO OPTIMAL FLOW CONTROL 1181
C. Douligeris and R. Mazumdar

INDEX OF AUTHORS

AN OPTIMUM SEQUENCE ESTIMATION APPROACH IN THE ABSENCE OF SYNCHRONIZATION

COSTAS N. GEORGHIADES
Electrical Engineering Department
Texas A&M University
College Station, TX 77843

ABSTRACT

A dynamic programming algorithm for joint maximum-likelihood estimation of sequences and symbol timings is introduced. The algorithm, which makes use of easily computed data, is easily implementable, extremely robust to timing-errors and alleviates, the need for a symbol synchronizer, which in general requires the complete sample-path to produce its timing estimates.

I. INTRODUCTION

In digital communication systems, an important and integral part of every practical receiver is the symbol synchronization subsystem. The function of the symbol synchronizer is to dictate to the receiver *where* in time the symbols are located. The receiver, subsequently, utilizes this timing information as if it were *perfect* in order to compute its decision statistics and decide *what* the transmitted symbols are. Clearly, this timing information is essential to the receiver and, if inaccurate, the effect could, and often is, catastrophic on system performance.

Extensive work on describing the performance degradation due to imperfect timing information has been done in the past and can be found, for example, in the excellent books of Lindsey and Simon [1] and Stiffler [2]. Results reported in these two references indicate that essentially disastrous results are obtained when the standard deviation of the timing jitter exceeds some small value. The severe performance degradation is due to the contamination of the decision statistics by adjacent symbols as a result of imperfect timing, -effectively reducing the signal-to-noise ratio- and because the receiver makes decisions assuming that the timing information is perfect. This assumption is in general not valid and leaves the receiver defenseless in the presence of, possibly, large timing-errors. We will see later in this paper that a receiver designed to make decisions in the presence of timing-errors has significant performance advantages over conventional receivers that rely blindly on the synchronization subsystem to provide timing information. Other approaches to the synchronization problem of interest can be found in [10].

II. ALGORITHM DERIVATION

In this section a dynamic programming algorithm for delay estimation and joint delay and sequence estimation is derived for a Gaussian channel and binary, antipodal signaling. Similar algorithms for other modulation schemes and for an optical Poisson channel can be found in [11].

We have the following communication problem

$$r(t) \;=\; \sum_{j=-\infty}^{\infty} S_j(t) \;+\; n(t), \tag{1}$$

where $S_j(t) \;\in\; \{S(t), -S(t)\}, \quad (j-1)T \leq t \leq jT$ is one of two antipodal modulation signals chosen independently and with equal probability and

$$\int_0^T S^2(t)dt \;=\; E.$$

Under our white Gaussian noise assumption, $n(t)$ is zero-mean, white Gaussian noise with spectral density $N_0/2$.

Not having a timing reference, the receiver computes sequentially the following statistics

$$r_j \;=\; \frac{1}{E}\int_{(j-1)T}^{jT} r(t-\tau')S(t)dt, \quad j = 1, 2, \cdots, N, \cdots \quad , \tag{2}$$

where, $\tau' \in [0, T)$ is the timing-error, as shown in Figure 1. Obviously, if $\tau' = 0$, the statistics $\{r_j, j = 1, 2, \cdots, N, \cdots\}$ constitute a sufficient statistic for the sequence estimation problem. In the presence of timing-error, however, although the r_j clearly contain information about the transmitted sequence, they are not a sufficient statistic and, thus, decisions based on them are not globally optimal. On the other hand, computing the statistics r_j requires very little complexity and, as we will see later, the performance obtained based on these statistics can be significantly better than the performance of conventional receivers utilizing a separate synchronizer.

The problem we address in this paper is that of optimally estimating sequences of symbols and the timing-error from the incomplete statistics $\{r_j, j = 1, 2, \cdots\}$. We begin by deriving a likelihood function of the observed data conditioned on the timing-error τ' and the modulation sequence.

Let $\mathbf{r} = (r_1, r_2, \cdots, r_N)$ be the vector containing the computed statistics $r_j, \; j = 1, 2, \cdots, N$ and $\mathbf{x} = (x_1, x_2, \cdots, x_{N+1})$, $x_i \in \{1, -1\}$, be the modulation sequence resulting in the received vector $\mathbf{r}$. The elements of $\mathbf{r}$ are conditionally independent, Gaussian random variables with mean

$$E[r_j/\mathbf{x}, \tau'] \;=\; x_j\frac{1}{E}\int_0^{\tau'} S(t)S(t+T-\tau')dt + x_{j+1}\frac{1}{E}\int_{\tau'}^{T} S(t)S(t-\tau')dt \;. \tag{3}$$

The variance of each r_j conditioned on the timing-error τ' and the modulation sequence $\mathbf{x}$ is

$$VAR[r_j/\mathbf{x}, \tau'] \;=\; N_0/2E, \tag{4}$$

independently of j. Assuming rectangular pulses $S(t)$ and letting

$$\tau \;=\; \frac{1}{E}\int_0^{\tau'} S(t)S(t+T-\tau')dt \;=\; \frac{\tau'}{T} \tag{5}$$

and $\nu_j = (x_j, x_{j+1})$, $\nu_j \in \{(-1,-1),(-1,1),(1,-1),(1,1)\}$ be the vector of modulation signals that affect the computation of r_j, we can express the mean of r_j by

$$\mu(\nu_j, \tau') \;=\; x_{j+1} \;+\; (x_j - x_{j+1})\tau \;. \tag{6}$$

Then, the conditional density of $\mathbf{r}$ given the timing error and the modulation sequence $\mathbf{x}$ is given by

$$p[\mathbf{r}/\mathbf{x}, \tau'] \;=\; \left(\frac{\pi N_0}{E}\right)^{-N/2} \exp\left\{-\frac{E}{N_0}\sum_{j=1}^{N}[r_j - \mu(\nu_j, \tau')]^2\right\} . \tag{7}$$

Taking logarithms on both sides and dropping terms that are not dependent on τ' or $\mathbf{x}$, we obtain the following likelihood function

$$\ell(\mathbf{r}/\mathbf{x}, \tau') \;=\; \sum_{j=1}^{N} r_j \mu(\nu_j, \tau') \;-\; \frac{1}{2}\sum_{j=1}^{N}\mu^2(\nu_j, \tau'). \tag{8}$$

An alternative statistic that indicates the dependence on τ' more explicitly is given by

$$\ell'(\mathbf{r}/\mathbf{x}, \tau') \;=\; \sum_{j=1}^{N} r_j[x_{j+1} + (x_j - x_{j+1})\tau] \;+\; \frac{1}{2}\tau[1-\tau]\sum_{j=1}^{N}(x_j - x_{j+1})^2 \;. \tag{9}$$

The two equivalent log-likelihood statistics in (8) and (9) are our starting point in deriving algorithms for synchronization and detection later.

Before we proceed further, we make the following important observation concerning the likelihood function (8) (or (9)). It can be shown easily that $\mu[(x_j, x_{j+1}), \tau'] = \mu[(x_{j+1}, x_j), (T - \tau')]$. This implies that a sequence $\mathbf{x}'$ for which $x'_j \;=\; x_{j+1}$ and $x'_{j+1} \;=\; x_j$ for all $j = 1, 2, \cdots$, will have exactly the same likelihood evaluation at $(T - \tau')$ as sequence $\mathbf{x}$ at τ', i.e., $\ell'[\mathbf{r}/\mathbf{x}, \tau'] = \ell'[\mathbf{r}/\mathbf{x}', (T - \tau')]$. Clearly, sequences that satisfy $x'_j \;=\; x_{j+1}$ and $x'_{j+1} \;=\; x_j$ for all $j = 1, 2, \cdots$, are sequences with alternating symbols. For these sequences, the receiver is incapable of distinguishing whether $\mathbf{x}$ or $-\mathbf{x}$ was sent from the observed data $\mathbf{r}$. In order to remove this binary ambiguity, we assume in Section III where performance is investigated that the receiver knows whether $\tau' < T/2$ or $\tau' > T/2$. This assumption should not be difficult to satisfy after the receiver observes some data, as we point out next.

a) Synchronization

Let us assume that in some magical way the modulation sequence is known. In practice this could be the case when a known preamble is sent for the purpose of symbol timing acquisition. The problem is to estimate optimally (in the sense of maximum-likelihood) the timing-error $p(\tau')$ from the observed incomplete data. Towards this end, we have the following necessary condition

$$\frac{\partial \ell'(\mathbf{r}/\mathbf{x}, \tau')}{\partial \tau} \;=\; \sum_{j=1}^{N} r_j(x_j - x_{j+1}) \;-\; [\tau - 1/2]\sum_{j=1}^{N}(x_j - x_{j+1})^2 \;=\; 0. \tag{10}$$

If the second summation in (10) is not zero, then the maximizing τ is given by

$$\hat{\tau}_N = \frac{1}{2} + \frac{\sum_{j=1}^{N} r_j(x_j - x_{j+1})}{\sum_{j=1}^{N}(x_j - x_{j+1})^2}. \tag{11}$$

A second derivative check on (10) verifies easily that (11) is indeed a maximum. Note that when the second sum in (10) *is* zero, then necessarily so is the first sum and by defining 0/0 to be zero, (11) can be used in this case. What the second summation in (10) being zero means is that the modulation sequence has no signal transitions, and thus, no timing information can be obtained from the observed signal. In this case, the optimal estimator in (11) in the absence of useful *a posteriori* data, uses its *a priori* information to produce its estimate.

For the general case of not necessarily rectangular signals $S(t)$, similar results can be derived, even though for notational simplicity we will restrict attention to rectangular pulses during the rest of the paper. The effect of the signal shape on synchronization enters through the correlations

$$r(T - \tau') = \int_0^{\tau'} S(t)S(t + T - \tau')dt$$

and

$$r(-\tau') = \int_{\tau'}^{T} S(t)S(t - \tau')dt. \tag{12}$$

Under the assumption that the sum in the denominator of (11) is not zero, it is easily shown that the estimates produced by equation (11), which assume that the modulation sequence is known, are unbiased. In other words,

$$E[\hat{\tau}_N/\mathbf{x}, \tau] = \tau. \tag{13}$$

Also, the variance of the normalized estimation error $e_N = (\tau - \hat{\tau}_N)$, given the modulation sequence $\mathbf{x}$ and the timing-error τ, is given by

$$VAR[e_N/\mathbf{x}, \tau] = \frac{1}{(2E/N_0)\sum_{j=1}^{N}(x_j - x_{j+1})^2}, \tag{14}$$

provided that the sum in the denominator is not zero. If it is zero, which is the case when the same symbol is transmitted repeatedly, then no timing information is conveyed by the data and the variance of the estimation error equals the *a priori* variance.

It is clear from (14) that the quality of the estimate is a function of the modulation sequence, with sequences with alternating symbols on one extreme providing the most timing information (smallest variance) and sequences of the same symbol being the worst. In general, the number of symbol changes determines the quality of the estimate. To show this explicitly, let the number of symbol changes in a sequence of length N

be D_N. Then the normalized estimation error variance in (14) can be expressed for $D_N \neq 0$ by

$$VAR(e_N/\mathbf{x}, \tau) \;=\; \frac{1}{4(2E/N_0)D_N}. \tag{15}$$

Equation (12) can be easily propagated recursively to yield the timing estimate [11].

It is clear that the assumption of a known modulation sequence makes the above algorithm rather impractical and only of academic interest. A way to make the algorithm practical at the cost of optimality is to use, instead of the actual modulation symbols, estimates of them provided by the receiver on a previous step. In this case the timing estimates will be lagging the symbol estimates by two symbol durations, which may not be unacceptable in situations where the timing jitter has a coherence time longer than two symbol durations. Denoting by $\hat{x}_j$ the estimated symbols and by τ^* the approximate timing estimate, the estimator in (12) is approximated by

$$\tau_N^* \;=\; \frac{1}{2} \;+\; \frac{\sum_{j=1}^{N} r_j(\hat{x}_j - \hat{x}_{j+1})}{\sum_{j=1}^{N}(\hat{x}_j - \hat{x}_{j+1})^2}. \tag{16}$$

Assuming that the modulation sequence is random and that the bit-errors are independent with a channel bit-error rate of q, it can be shown in Appendix B that the bias of the estimate and the variance of the estimation error $e_N^* \;=\; (\tau_N^* - \tau)$ are given for large N by

$$E[\tau_N^*/\tau] \;=\; \tau(1-2q) \;+\; q \tag{17}$$

$$VAR(e_N^*) \leq \frac{1}{2(2E/N_0)N} \;+\; \frac{2q(1-q)}{N}. \tag{18}$$

Clearly, the estimation error variance can be made arbitrarily small by observing longer sequences. On the other hand, the estimation bias is only a function of the bit-error rate q and cannot be reduced by observing longer sequences. If the bit-error rate can be accurately estimated, then the bias can be eliminated if necessary, or it can be ignored for systems with low bit-error rate q. The point of the above derivations is that by using estimated symbols instead of actual symbols that are not known *a priori*, it may be possible to obtain a timing estimate that, for large N and small q, approaches the optimal estimate. Even if not optimal, estimates produced by (17) should be sufficient in determining whether $\tau' > 1/2$ or $\tau' < 1/2$. As we noted earlier, this assumption is made in the next section when evaluating the performance of the algorithm introduced next.

b) Detection

We now turn our attention to the problem of estimating a sequence of symbols from the incomplete data $\mathbf{r}$. Again, our starting point is the likelihood equation (8) (or (9)). Our approach is to jointly optimize (8) with respect to the timing-error τ and the modulation sequence $\mathbf{x}$. One way this can be done is by inserting in (8) the optimal timing estimate $\hat{\tau}$ of (12). The resulting likelihood function, already optimized with respect to τ becomes

$$\ell(\mathbf{r}/\mathbf{x}, \hat{\tau}) \;=\; \sum_{j=1}^{N} r_j \mu(\nu_j, \hat{\tau}) \;-\; \frac{1}{2}\sum_{j=1}^{N} \mu^2(\nu_j, \hat{\tau}), \tag{19}$$

with $\hat{\tau}$ given by (12). Using the necessary condition (10) in the likelihood function (9), we obtain the following expression

$$\ell(\mathbf{r}/\mathbf{x}, \hat{\tau}) = \sum_{j=1}^{N} r_j x_{j+1} + \frac{\hat{\tau}^2}{2} \sum_{j=1}^{N} (x_j - x_{j+1})^2, \tag{20}$$

with $\hat{\tau}$ given by (12).

Equation (20) can be implemented efficiently through a dynamic programming algorithm, by observing that it can be propagated recursively and that at the $j-th$ step, the incremental "cost" depends only on the observed statistic r_j and the state $\nu_j = (x_j, x_{j+1})$. This observation leads to the state diagram in Figure 2 and the corresponding trellis diagram in Figure 3 which describe the sequence of states visited as a function of the modulation sequence $\mathbf{x}$. Notice that in the above diagrams the logical level 0 corresponds to the physical level -1 and the logical level 1 corresponds to the physical level 1 for binary PSK.

A major disadvantage of the above algorithm, which is the reason another algorithm is introduced next, is that there is no obvious way the knowledge that $\tau < 1/2$ or $\tau > 1/2$ can be incorporated into the algorithm. Thus, it may not be possible to remove the binary ambiguity when sequences with alternating symbols are transmitted. Also, it is quite possible that the estimates produced by (29) lie outside the interval $[0, 1)$. In this case also, the question of how estimates outside the range $[0, 1)$ are resolved at intermediate steps in the algorithm has no obvious answer. It is possible that branches eliminated at intermediate steps when the constraint on $\hat{\tau}$ is not imposed may eliminate sequences that are better candidates for decoding when the constraint on $\hat{\tau}$ is taken into account. Since there is no easy way that this constraint can be satisfied causally, an implementation of (20) may require the evaluation of 2^{N+1} statistics before a decision is made, a totally impractical prospect.

We now turn our attention to a second approach in optimizing (8) (or (9)), which has significant computational advantages over the above method.

The approach is simple: quantize the parameter $\tau \in [0, 1)$ into a discrete number, M, of values from some set $\mathcal{T} = \{t_1, t_2, \cdots, t_M\}$. Then use a dynamic programming (Viterbi) algorithm to propagate the likelihood function in (8) for each of the possible values of $\tau \in \mathcal{T}$. Although the quantization is bound to introduce some performance degradation, it is expected that the degradation can be made small by choosing M to be sufficiently large. Results obtained indicate that very little performance degradation is incurred for values of M exceeding some small threshold value.

It is clear that the maximization of (8) when τ takes values from a finite set can be implemented efficiently through a dynamic programming (Viterbi) algorithm, where at each step in the algorithm a vector of M likelihood statistics is propagated. Figures 2 and 3 show the state and trellis diagrams respectively.

Let $\mathcal{L}^J = \{\ell_1^J, \ell_2^J, \cdots, \ell_M^J\}$ be the vector of propagated statistics at step J where

$$\ell_i^J = \ell(\mathbf{r}/\mathbf{x}_J, t_i) \tag{21}$$

is the likelihood statistic corresponding to the $i-th$ quantization level t_i and, as above,

$\mathbf{x}_J = (x_1, x_2, \cdots, x_{J+1})$. Then each element of $\mathcal{L}$ can be propagated recursively using

$$\ell_i^J = \ell_i^{J-1} + \left[r_J \mu(\nu_J, t_i) - \frac{1}{2}\mu(\nu_J, t_i)^2 \right], \tag{22}$$

where the four possible $\mu(\nu_J, t_i)$ corresponding to the four states are

$$\begin{aligned} \mu[(-1,-1), t_i] &= -1 \\ \mu[(1,1), t_i] &= -\mu[(-1,-1), t_i] = 1 \\ \mu[(-1,1), t_i] &= 1 - 2t_i \\ \mu[(1,-1), t_i] &= -\mu[(-1,1), t_i] = -1 + 2t_i. \end{aligned} \tag{23}$$

Notice that the computations for each value of t_i are completely independent from computations for other quantization times and that they can be performed in parallel at each step. Also, the incremental path metric does not depend on the branch but only on the destination state, which simplifies computations even further. At each step in the trellis, $4M$ statistics are computed corresponding to the surviving paths. Both the statistics corresponding to surviving paths as well as the surviving paths are stored at each step. At some predefined depth N in the trellis, a sequence decision is made by choosing the largest statistic over all states and quantization times t_i and releasing the corresponding survivor sequence.

III. PERFORMANCE-CONCLUSION

In this section we present simulation results that compare the performance of the dynamic programming algorithm described above for the binary PSK Gaussian channel, to that of conventional receivers. More results can be found in [11].

Figure 4 compares the performance of the introduced algorithm to that of a conventional receiver utilizing a separate synchronizer with a steady-state timing-error described by a Tichonov density. The performance of the conventional receiver was derived analytically from results in [1] and is plotted for three values of the standard deviation of the normalized timing-error, $\sigma = 0.08,\ 0.1$, and 0.2. The performance of the introduced algorithm is plotted for $N = 30$ and for both a uniform timing-error and a constant normalized timing-error equal to 0.5. Even in this extremely pessimistic scenario, the overwhelming superiority of the introduced algorithm over the conventional receiver is obvious in the region of signal-to-noise ratios of interest, corresponding to error probabilities of less than 10^{-4}. This improvement is expected to be even more significant as the signal-to-noise ratio is increased.

The robustness and performance of the introduced algorithm, as seen from Figure 4, indicate that it is possible to built receivers that do not utilize separately designed symbol synchronizers and make decisions from observations of easily computed statistics. Besides the performance advantage shown above, the introduced algorithm has the extra merit that it is easily implemented digitally, in contrast to symbol synchronizers that require in general the complete received waveform to produce their timing estimates. The little complexity introduced by the trellis search should be easily offset by the stated advantages of the algorithm.

IV. REFERENCES

1. W.C. Lindsey and M.K. Simon, *Telecommunications Systems Engineering,* Prentice-Hall, Inc., Englewood Cliffs, N.J., 1973.
2. J.J Stiffler, *Theory of Synchronous Communications*, Prentice Hall Inc., Englewood Cliffs, N.J., 1971.
3. H. Kobayashi, "Simultaneous adaptive estimation and decision algorithm for carrier modulated data transmission system," IEEE Transactions on Communication Technology, vol. COM-19, pp. 268-280, June 1971.
4. D.D. Falconer and J. Salz, "Optimal reception of digital data over the Gaussian channel with unknown delay and phase jitter," IEEE Transactions on Information Theory, vol. IT-23, pp. 117-126, January 1977.
5. Marc Moeneclaey, "Synchronization problems in PAM systems," IEEE Transactions on Communications, vol. COM-28, No. 8, pp. 1130-1136, August 1980.
6. M.K. Simon, "Further results on optimum receiver structures for digital phase and amplitude modulated signals," International Conference on Communications, Toronto, Canada 1978.
7. H.H. Meyers and L.E. Franks, "Joint carrier phase and symbol timing for PAM systems," IEEE Transactions on Communications, vol. COM-28, No. 8, pp. 1121-1129, August 1980.
8. U. Mengali, "Synchronization of QAM signals in the presence of ISI," IEEE Transactions on Aerospace and Electronic Systems, vol. AES-12, pp. 556-560, September 1976.
9. C.N. Georghiades, "Some Optimal and Suboptimal Receivers Deriving Information from Nonsynchronized Poisson Data," to appear in the IEEE Transactions on Information Theory.
10. M.P. Advani and C.N. Georghiades, "Jointly Optimal Receivers for the Optical Pulse-Position Modulation Channel," under review by the IEEE Transactions on Communications.
11. C.N. Georghiades, "Optimum Delay and Sequence Estimation from Incomplete Data," under review by the IEEE Transactions on Information Theory.

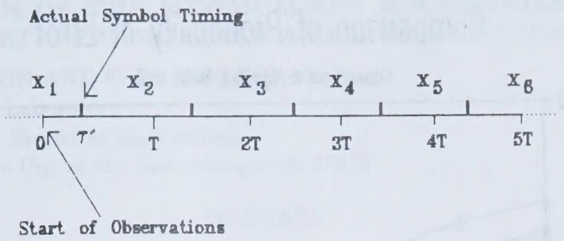

Actual Symbol Timing
x_1
x_2
x_3
x_4
x_5
x_6
0
τ'
T
2T
3T
4T
5T
Start of Observations

Figure 1. Illustration of the Timing-Error τ'

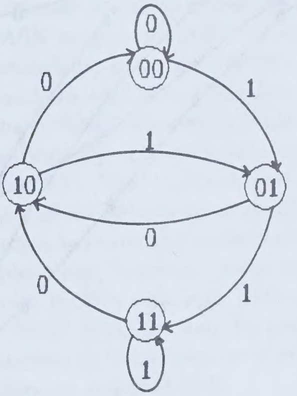

0
00
0
1
1
10
01
0
0
1
11
1

Figure 2. State diagram for binary systems.

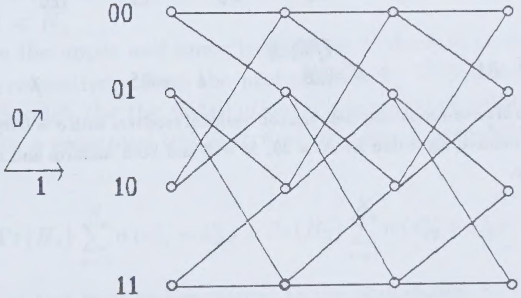

00
01
10
11
0
1

Figure 3. Trellis for binary systems.

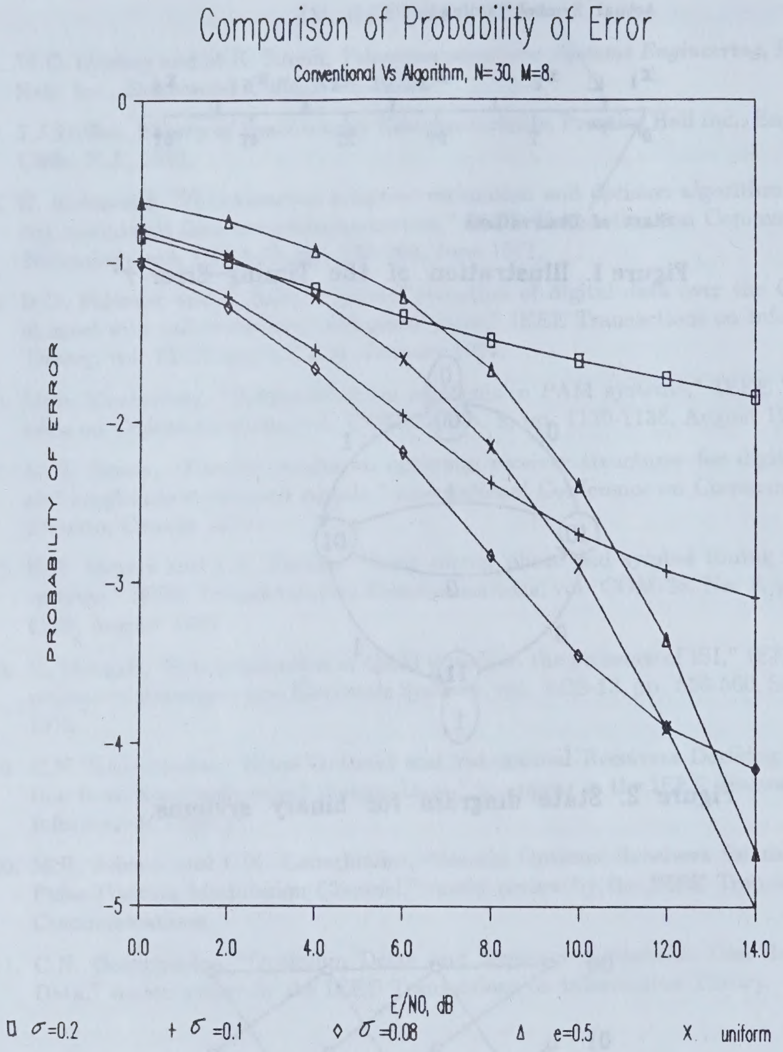


Fig. Comparison of error-probability between conventional receivers with $\sigma = 0.08,\ 0.1,\ 0.2$ and the introduced algorithm for $N = 30$, $M = 8$ and both uniform and maximum timing-error.

COMPUTATION OF THE MINIMUM-ASN BOUNDARIES FOR THE MODIFIED SPRT: PERFORMANCE ANALYSIS AND COMPARISONS

Ö. SNORRASON AND F. D. GARBER
ElectroScience Laboratory
Department of Electrical Engineering
The Ohio State University, Columbus, Ohio 43210

SUMMARY

This paper is concerned with the computation of the optimum boundaries, in the sense of minimizing the average sample number (ASN) for a given set of error probabilities and a limited number of samples N, for sample-dependent boundary forms of the Wald SPRT. Of particular interest are cases where the maximum sample number N is small and cases where the ASN is approximately the same size as N.

The binary sequential probability ratio test (SPRT) developed by Wald *et. al.* [1] is a powerful technique that has been applied to a number of decision systems. In addition to the specification of the basic SPRT algorithm, Wald derived useful design formulas, known as Wald's approximations [2], that relate the maximum error probabilities and the average sample number (ASN). The significance of these approximations is demonstrated by their independence of the underlying probability distributions as well as their independence of the correlation between the individual measurements.

For most practical applications, however, some means of limiting the maximum number N of allowed samples is often required. This restriction may take the form of a truncated version of the Wald SPRT or may be realized by modifying the decision boundaries so as to be a function of the number n of samples for $n \leq N$, while ensuring that no more than N samples are required [3,4].

A noted characteristic of Wald's approximations that is common to all forms of SPRT's is that they are most accurate when the error probabilities are very small and the required ASN is very large. When these conditions are not satisfied, as is the case for many practical situations, the effect of neglecting the excess of the probability ratio over the decision boundaries becomes significant and the results obtained from the approximations may be misleading. The application of Wald's approximations to forms of limited sample test is only valid when the effect of neglecting the excess is negligible and when ASN $\ll N$.

If we denote the upper and lower boundaries of the test at stage n, $1 \leq n \leq N$ by $a(n)$ and $b(n)$, respectively, then the probabilities, $e_{ij}^{n} \triangleq \Pr\{\text{decide } H_i \text{ given } H_j\}$, are characterized through the the distribution of accumulated value of decision statistics [5]. The ASN for a sequential test described by a set of probabilities, $\{e_{ij}^{n}, 1 \leq n \leq N\}$ is given by

$$E\{n\} = \Pr\{H_1\} \sum_{n=1}^{N} n\,(e_{11}^{n} + e_{21}^{n}) + \Pr\{H_2\} \sum_{n=1}^{N} n\,(e_{22}^{n} + e_{12}^{n})$$

The optimum test may be formulated as the boundaries $a(n)$ and $b(n)$, $1 \leq n \leq N$, that minimize $E\{n\}$ subject to the design constraints:

Table 1: ASN for various values of N with $e_{12} = e_{21}$ for the Optimum boundaries, Wald's truncated SPRT and Fu's modified boundaries.

		Optimum	Wald	Fu, $r = 0.5$	Fu, $r = 1$	Fu, $r = 2$
$N = 3$	$e_{12} = 0.1$	1.236	1.237			
$N = 3$	$e_{12} = 0.045$	1.829	2.003	1.837	1.844	2.017
$N = 3$	$e_{12} = 0.04165$	1.911	2.659	1.928	1.921	2.080
$N = 5$	$e_{12} = 0.1$	1.234	1.234			
$N = 5$	$e_{12} = 0.02$	2.222	2.320	2.227	2.258	2.344
$N = 5$	$e_{12} = 0.013$	2.466	3.167	2.568	2.564	2.636
$N = 7$	$e_{12} = 0.013$	2.404	2.458		2.453	
$N = 7$	$e_{12} = 0.005$	2.820	3.332	3.035	3.065	3.294
$N = 10$	$e_{12} = 0.001$	3.605	4.129	3.860	3.988	4.446

1. $e_{12} = \sum_{n=1}^{N} e_{12}^{n}$,
2. $e_{21} = \sum_{n=1}^{N} e_{21}^{n}$,
3. $a(N) = b(N)$,

where the first two relations determine the average error probabilities of the test and the third constraint imposes a maximum on the number of samples allowed.

In general, it appears that the derivation of explicit formulas for the optimum sets of boundaries is extremely difficult. However, given knowledge of the underlying probability distributions, the optimum solution can be obtained through the use of numerical gradient methods or nonlinear programming techniques.

Table 1 shows preliminary results for the reduction of the ASN that can be achieved through the use of the optimum boundaries. In this table, the ASN required by the optimum test is compared with that required by Wald's truncated SPRT and the SPRT that uses Fu's modified boundaries. From these results, it is clear the reduction in the ASN achieved by optimizing the decision boundaries is most significant when the resulting average error probability is small.

REFERENCES

[1] A. Wald, *Sequential Analysis.* New York: Wiley, 1947.

[2] H. V. Poor, *An Introduction to Signal Detection and Estimation.* New York: Springer-Verlag, 1988.

[3] Y. T. Chien and K. S. Fu, "A modified sequential recognition machine using time-varying stopping boundaries," *IEEE Transactions on Information Theory*, vol. IT–12, no. 2, pp. 201–214, April 1966.

[4] J. J. Bussgang and M. B. Marcus, "Truncated sequential hypothesis tests," *IEEE Transactions on Information Theory*, vol. IT–13, no. 3, pp. 512–516, July 1967.

[5] B. A. Rozanov, "Distribution of accumulated value of decision statistics in sequential analysis," *Radio Engineering Electronic Physics*, vol. 17, pp. 1667–1673, 1972.

A PARAMETRIC FORM FOR SIEVE-CONSTRAINED MAXIMUM-LIKELIHOOD DENSITY ESTIMATION FROM NOISY DATA*

TIMOTHY J. SCHULZ and DONALD L. SNYDER
Electronic Systems and Signals Research Laboratory
Department of Electrical Engineering
Washington University
St. Louis, MO 63130

Summary

Nonparametric unconstrained maximum-likelihood estimation of probability densities from noisy measurements is a fundamentally unstable problem. This instability presents itself in the form of estimates which are very rough, exhibiting high peaks and low valleys throughout. This effect is not unique to density estimation as it has also been observed in estimation of point process intensities [1]. In [1] and [2], Grenander's method of sieves is suggested as an approach to stabalize estimates of point process intensities obtained from noisy observations of the point locations. As a result of the similarities between point process intensities and probability densities, the method of sieves has been suggested for probability density estimation from noisy data [3]. In this paper, we extend results of Geman and Huang [4] and Geman [5] to show that sieve-constrained maximum-likelihood estimates of probability densities from noisy measurements are parameterized by at most 2N parameters when a Gaussian-kernel sieve is used and measurement errors are normally distributed, where N is the number of measurements. We present experimental results indicating that the number of distinct parameters may be significantly less than 2N.

In our model, we are presented with measurement data of the form $y=x+w$, where x is the random variable whose density is to be estimated, w is an independent Gaussian random variable with zero mean and known variance, and y is the observed random variable. We are presented with a series of N independent measurements $y_1, y_2, \dots, y_N$ and are to estimate the probability density $p(x)$ of x. The loglikelihood of these data, as a functional of the unknown density, is given by

$$L(p) = \sum_{n=1}^{N} \ln \left[\int_X p_y(y_n|x) p(x) dx \right], \tag{1}$$

where $p_y(y|x)$ is the conditional density for y given x and has the form

$$p_y(y|x) = \frac{1}{\sqrt{2\pi\sigma_w^2}} e^{\frac{-(y-x)^2}{2\sigma_w^2}}, \tag{2}$$

where σ_w^2 is the variance of the noise term w. Equation (1) may be written alternatively as

$$L(p) = \int_Y \ln \left[\int_X p_y(y|x) p(x) dx \right] N(dy), \tag{3}$$

where $N(A)$ is the number of data samples in set A. Unconstrained maximization of (3) leads to unstable estimates of p, which motivates the introduction of a constrained subspace over which the maximization is performed. The use of a kernel sieve as a constraint results in maximizing L subject to the estimate of p being a member of the sieve set

$$S_m = \left\{ p : p(x) = \int_Z s_m(x|z) dF(z) \right\}, \tag{4}$$

where $s_m(x|z)$ is the kernel of the sieve, and F is an arbitrary probability distribution. The form of the kernel s_m considered here is a Gaussian density function such that

---

*This work was supported by the National Science Foundation under grant number MIP-8722463

$$s_m(x|z) = \frac{1}{\sqrt{2\pi\sigma_m^2}} e^{\frac{-(x-z)^2}{2\sigma_m^2}} , \tag{5}$$

with σ_m specifying the width of the kernel. Maximizing (3) subject to $p \in S_m$ is equivalent to maximizing the following funtional of F

$$L(F) = \int_Y \ln \left[\int_Z k(y|z) dF(z) \right] N(dy) , \tag{6}$$

where k is a Gaussian kernel with zero mean and variance $\sigma_r^2 + \sigma_m^2$. After the F which maximizes (6) is found, $\hat{p}$ is obtained as

$$\hat{p}(x) = \int_Z s_m(x|z) dF(z) . \tag{7}$$

Slightly extending results of Geman [5], the F maximizing (6) is a discontinuous distribution having the parametric form

$$F(z) = \sum_{j=1}^{N} \alpha_j u(z - \mu_j) , \tag{8}$$

where $u(z)$ is the unit-step function, which results in p having the parametric form

$$p(x) = \sum_{j=1}^{N} \alpha_j s_m(x|\mu_j) . \tag{9}$$

Thus, the maximum-likelihood estimate consists of a linear superposition of at most N Gaussian kernels of the sieve having undetermined scale and location parameters, $\{\alpha_j\}$ and $\{\mu_j\}$, respectively. This result can readily be extended to include a resolution kernel to suppress Gibb's overshoot, as discussed in [2,3].

A numerical procedure based on the expectation-maximization algorithm is given for producing numerical estimates of the parameters in (8) and (9). One possibility is that the use of the parameterized form may alleviate the Gibb's overshoot seen when using expectation-maximization algorithms derived without knowledge of the parameterized form of the estimate [2,3]. Comparisons made between constrained maximum-likelihood estimates produced without using the parametric form of the estimate to those produced using the parametric form indicate that the estimates produced are virtually identical. Thus, our preliminary conclusion is that the parametric form does not appear to offer improvements in resolution.

REFERENCES

[1] D. L. Snyder and M. I. Miller, "The Use of Sieves to Stabilize Images Produced with the EM Algorithm for Emission Tomography," IEEE Trans. on Nuclear Science, Vol. NS-32, No. 5, pp. 3864-3872, Oct. 1985.

[2] D. L. Snyder, M. I. Miller, L. J. Thomas, Jr., and D. G. Politte, "Noise and Edge Artifacts in Maximum-Likelihood Reconstructions for Emission Tomography," IEEE Trans. on Medical Imaging, Vol. MI-6, No. 3, pp. 228-238, Sept. 1987.

[3] D. L. Snyder, M. I. Miller, and T. J. Schulz, "Constrained Probability-Density Estimation from Noisy Data," Proceedings of the 1988 Conference on Information Sciences and Systems, Princeton University, Princeton, New Jersey, pp. 170-172, March 1988.

[4] S. Geman and C. Hwang, "Nonparametric Maximum Likelihood Estimation by the Method of Sieves," The Annals of Statistics, Vol. 10, pp. 401-414, 1982.

[5] S. Geman, "Sieves for Nonparametric Estimation of Densities and Regressions," Repts. in Pattern Analysis, No. 99, D.A.M., Brown Universtity, 1981.

ASYMPTOTICALLY EFFICIENT ESTIMATION OF PRIOR PROBABILITIES IN MULTICLASS FINITE MIXTURES

G. R. DATTATREYA
Department of Computer Science
University of Texas at Dallas
P.O. Box 688, Richardson, TX 75083

and

L. N. KANAL
Department of Computer Science
University of Maryland
College Park, MD 20742

Abstract: Efficient estimators do not exist for mixing probabilities in finite class mixtures. Earlier published work on this topic has merely conjectured that asymptotically efficient estimators exist. In this paper, we synthesize a prior probability estimator, a candidate for asymptotic efficiency, from within the class of recursive estimators proposed in our earlier work [2]. We prove convergence with probability one and asymptotic efficiency by invoking the multidimensional stochastic approximation theorem and a theorem on the asymptotic distributions of stochastic approximation procedures. The estimator can be implemented in practice for continuous, discrete, and mixed class conditional density functions, although continuous and mixed densities generally require repeated evaluation of expectations of certain functions through numerical techniques. Results of a simulation experiment with discrete densities are discussed. Variations of the estimator, for computational simplicity, are discussed.

I ESTIMATION PROBLEM AND CONDITIONS FOR EFFICIENCY

We observe a vector sequence $\mathbf{X}^n = \{\mathbf{X}_1, \ldots, \mathbf{X}_n\}$ of independent identically distributed (iid) random variables, each with the density function

$$p(\mathbf{x}) = \sum_{l=1}^{M} P_l \, p(\mathbf{x}|\omega_l) \tag{1}$$

where $p(\mathbf{x}|\omega_l)$, $l=1, \ldots, M$ are the M class conditional density functions and P_l, $l=1, \ldots, M$ are the M prior probabilities whose estimation we are concerned with. Since $\sum P_l = 1$, we can effectively consider P_l, $l=1, \ldots, M-1$ as the unknown parameters of $p(\mathbf{x})$. Therefore,

$$p(\mathbf{x}) = p(\mathbf{x}|\omega_M) + \sum_{l=1}^{M-1} P_l \left[p(\mathbf{x}|\omega_l) - p(\mathbf{x}|\omega_M) \right] \tag{2}$$

The joint density of $\mathbf{X}^n$ is

$$p(\mathbf{x}^n) = \prod_{k=1}^{n} \left\{ p(\mathbf{x}_k|\omega_M) + \sum_{l=1}^{M-1} P_l \left[p(\mathbf{x}_k|\omega_i) - p(\mathbf{x}_k|\omega_M) \right] \right\}. \tag{3}$$

We will use $M-1$ functions of observations. These are denoted by $h_i(\mathbf{x})$, $i=1, \ldots, M-1$. Their expectations are $h_{ij} = E[h_i(\mathbf{X})|\omega_j]$. If $h_i(\mathbf{x})$ are functions of the current prior probability estimates (PPEs), they are denoted by $h_i(\mathbf{x}_n, \mathbf{P}^*(n-1))$.

$$h_{ij}(\mathbf{P}^*(n-1)) = E[h_i(\mathbf{X}, \mathbf{P}^*(n-1))|\omega_j].$$

$\mathbf{P}^*(n-1)$ will be defined as a simple function of $\hat{\mathbf{P}}(n-1)$, to ensure that the former is a valid probability vector. The sample average of the first of the above functions of observations is denoted by

$$\hat{h_i}(n) = \frac{1}{n} \sum_{k=1}^{n} h_i(\mathbf{x}_k).$$

A matrix $\mathbf{H}$ composed of elements $H_{ij} = h_{ij} - h_{iM}$ will be used. Similarly,

$$H_{ij}(\mathbf{P}^*(n-1)) = h_{ij}(\mathbf{P}^*(n-1)) - h_{iM}(\mathbf{P}^*(n-1))$$

and $\mathbf{h}_M$ is a column vector containing $h_{1M}, \ldots, h_{M-1M}$.

A. Estimation Equations

The class of estimators under study is derived as follows. Take the expectations of the functions $h_i(\mathbf{X})$, $i=1, \ldots, M-1$ and rearrange.

$$\sum_{j=1}^{M-1}(h_{ij} - h_{iM})P_j = E[h_i(\mathbf{X})] - h_{iM}, \quad i = 1, \ldots, M-1. \tag{4}$$

h_{ij}, $j=1, \ldots, M$, $i=1, \ldots, M-1$ are known; $E[h_i(\mathbf{X})]$, $i=1, \ldots, M-1$ can be estimated as sample averages of the functions $h_i(\mathbf{x}_k)$ of observations. Thus the matrix estimation equation is

$$\hat{\mathbf{P}}(n) = \mathbf{H}^{-1}[\hat{\mathbf{h}}(n) - \mathbf{h}_M]. \tag{5}$$

Manipulating (5), an alternative form of the matrix estimator, directly suitable for recursive computation is obtained.

$$\hat{\mathbf{P}}(n) = \frac{n-1}{n}\hat{\mathbf{P}}(n-1) + \frac{1}{n}\mathbf{H}^{-1}[\mathbf{h}(\mathbf{x}_n) - \mathbf{h}_M]. \tag{6}$$

If $\mathbf{h}(\mathbf{X})$ and hence $\mathbf{H}$ use functions of the existing estimate $\hat{\mathbf{P}}(n-1)$, they vary from sample to sample. This dependence can be explicitly incorporated into (6) as

$$\hat{\mathbf{P}}(n) = \frac{n-1}{n}\hat{\mathbf{P}}(n-1) + \frac{1}{n}\mathbf{H}^{-1}(\mathbf{P}^*(n-1))\Big[\mathbf{h}(\mathbf{x}_n, \mathbf{P}^*(n-1)) - \mathbf{h}_M(\mathbf{P}^*(n-1))\Big]. \tag{7}$$

B. Assumptions

(1) $0 < P_l < 1$, $l = 1, \ldots, M$.

(2) Prior probabilities are identifiable. This implies [2] that the class conditional density functions are linearly independent functions.

(3) Class conditional mean vectors possess finite components. Class conditional density functions can possess Dirac impulse function components. Therefore, integrations for carrying out expectations should be interpreted appropriately. We may not be able to evaluate the density function values at all points due to these impulse functions.

C. Conditions for Efficiency

Variances of unbiased estimates of parameters of a distribution function are bounded from below by the Cramer–Rao minimum variance bounds (MVB) provided the distribution function satisfies the *Regularity Conditions* [8]. These bounds are $\mathrm{Var}[\hat{P}_i(n)] \geq J^{ii}$ where J^{ii} is the ii-th element of the inverse of the Fischer's information matrix $\mathbf{J}$ whose elements are given by

$$J_{ij} = -E\left[\frac{\partial^2 \ln p(\mathbf{x}^n)}{\partial P_i \partial P_j}\right].$$

If the variances equal J^{ii}, the estimate is called an efficient estimate. An efficient estimate, if exists, is necessarily of the form

$$\hat{P}_i(n) = P_i + \sum_{j=1}^{M-1} f_{ij}(\mathbf{P}) \frac{\partial \ln p(\mathbf{x}^n)}{\partial P_j} \tag{8}$$

for all values of $\mathbf{x}^n$. (8) provides us with a method to test whether or not a given estimator is efficient. With respect to our estimation problem, $p(\mathbf{x}^n)$ is an nth degree polynomial in $P_1, \ldots, P_{M-1}$. Therefore, its partial derivatives exist and are absolutely integrable for any finite P_l, verifying the Regularity conditions.

$$\frac{\partial \ln p(\mathbf{x}^n)}{\partial P_i} = \sum_{k=1}^{n} \frac{p(\mathbf{x}_k|\omega_i) - p(\mathbf{x}_k|\omega_M)}{p(\mathbf{x}_k|\omega_M) + \sum_{l=1}^{M-1} P_l[p(\mathbf{x}_k|\omega_l) - p(\mathbf{x}_k|\omega_M)]}, \tag{9}$$

$$\frac{\partial^2 \ln p(\mathbf{x}^n)}{\partial P_i \partial P_j} = -\sum_{k=1}^{n} \frac{[p(\mathbf{x}_k|\omega_i) - p(\mathbf{x}_k|\omega_M)][p(\mathbf{x}_k|\omega_j) - p(\mathbf{x}_k|\omega_M)]}{\left[p(\mathbf{x}_k|\omega_M) + \sum_{l=1}^{M-1} P_l[p(\mathbf{x}_k|\omega_l) - p(\mathbf{x}_k|\omega_M)]\right]^2}. \tag{10}$$

$$J_{ij} = -E\left[\frac{\partial^2 \ln p(\mathbf{X}^n)}{\partial P_i \partial_j P_j}\right] = \sum_{k=1}^{n} \int \frac{[p(\mathbf{x}_k|\omega_i) - p(\mathbf{x}_k|\omega_M)][p(\mathbf{x}_k|\omega_j) - p(\mathbf{x}_k|\omega_M)]}{p(\mathbf{x}_k|\omega_M) + \sum_{l=1}^{M-1} P_l[p(\mathbf{x}_k|\omega_l) - p(\mathbf{x}_k|\omega_M)]} dx$$

$$= n \int \frac{[p(\mathbf{x}|\omega_i) - p(\mathbf{x}|\omega_M)][p(\mathbf{x}|\omega_j) - p(\mathbf{x}|\omega_M)]}{p(\mathbf{x}|\omega_M) + \sum_{l=1}^{M-1} P_l[p(\mathbf{x}|\omega_l) - p(\mathbf{x}|\omega_M)]} d\mathbf{x}. \tag{11}$$

Appendix I lists properties of several functions used in this paper, including the invertibility of $\mathbf{J}$. Let us examine the form of a prospective efficient estimator. (8) takes the form

$$\hat{P}_i(n) = P_i + \sum_{j=1}^{M-1} \sum_{k=1}^{n} f_{ij}(\mathbf{P}) \frac{p(\mathbf{x}_k|\omega_j) - p(\mathbf{x}_k|\omega_M)}{p(\mathbf{x}_k|\omega_M) + \sum_{l=1}^{M-1} P_l[p(\mathbf{x}_k|\omega_l) - p(\mathbf{x}_k|\omega_M)]}. \tag{12}$$

Therefore, an efficient estimator of $\mathbf{P}$ requires P_l, the components of the vector $\mathbf{P}$. This shows the nonexistence of efficient PPEs, for general $p(\mathbf{x}|\omega_i)$.

III. MINIMIZATION OF VARIANCE

The expression for J_{ij} in (11) suggests the examination of

$$h_i(\mathbf{x}) = \frac{p(\mathbf{x}|\omega_i) - p(\mathbf{x}|\omega_M)}{p(\mathbf{x}|\omega_M) + \sum_{l=1}^{M-1} P_l[p(\mathbf{x}|\omega_l) - p(\mathbf{x}|\omega_M)]} \tag{13}$$

as a candidate for minimizing the variance of the estimate. With $h_i(\mathbf{x})$ given by (13), we have

$$E[h_i(\mathbf{X})] = 0 \text{ and} \tag{14}$$

$$E[h_i(\mathbf{X})h_j(\mathbf{X})] = \int \frac{[p(\mathbf{x}|\omega_i) - p(\mathbf{x}|\omega_M)][p(\mathbf{x}|\omega_j) - p(\mathbf{x}|\omega_M)]}{p(\mathbf{x}|\omega_M) + \sum_{l=1}^{M-1} P_l[p(\mathbf{x}|\omega_l) - p(\mathbf{x}|\omega_M)]} \, d\mathbf{x} = \frac{1}{n} J_{ij}. \tag{15}$$

From (14) and (15), $E[\mathbf{h}(\mathbf{X})] = 0$, $\mathbf{Cov}[\mathbf{h}(\mathbf{X})] = \frac{1}{n}\mathbf{J}$.

$$H_{ij} = E[h_i(\mathbf{X})|\omega_j] - E[h_i(\mathbf{X})|\omega_M] = \frac{1}{n} J_{ij}. \tag{16}$$

(16) follows from (11). Note that $\mathbf{H}$ in (11) is symmetric. $\mathbf{H}$ with $\mathbf{h}(\mathbf{X})$ defined by (13), and $\mathbf{J}$ are invertible (see Appendix I). From (5), (14), (15) and (16),

$$\mathbf{Cov}[\hat{\mathbf{P}}(n)] = \frac{1}{n}\{\frac{1}{n}\mathbf{J}\}^{-1}\{\frac{1}{n}\mathbf{J}\}\{\frac{1}{n}\mathbf{J}\}^{-1} = \mathbf{J}^{-1} \quad \text{and} \tag{17}$$

$$\mathrm{Var}[P_i(n)] = J^{ii}. \tag{18}$$

Thus, the variances of the components of the hypothetical estimator $\hat{\mathbf{P}}(n) = \mathbf{H}^{-1}[\hat{\mathbf{h}}(n) - \mathbf{h}_M]$ with $h_i(\mathbf{x})$ given by (13) equal the MVB components. From Appendix I, given $\mathbf{P}$, the above function can be evaluated for all finite $\mathbf{x}$ even when class conditional density functions possess Dirac impulse functions. A practical way to utilize the favorable properties of the above estimator is by using $\hat{\mathbf{P}}(n-1)$ instead of the actual, unknown $\mathbf{P}$ on the RHS of (13). The next section is concerned with the asymptotic properties of such an estimator.

IV. ASYMPTOTIC EFFICIENCY

We are interested in analyzing the estimator with

$$0 < \hat{P}_l(0)\,, \quad l = 1, \ldots, M-1 \quad \text{and} \quad \sum_{l=1}^{M-1} \hat{P}_l(0) < 1; \tag{19}$$

$$\begin{aligned} \mathbf{P}^*(n) &= \hat{\mathbf{P}}(n), \quad \text{if } 0 < \hat{P}_l(n),\ l = 1, \ldots, M-1 \text{ and } \sum_{l=1}^{M-1} \hat{P}_l(n) < 1, \\ &= \mathbf{P}^*(n-1)\,, \quad \text{otherwise,} \end{aligned} \tag{20}$$

$$h_i(\mathbf{x}_n, \mathbf{P}^*(n-1)) = \frac{p(\mathbf{x}_n|\omega_i) - p(\mathbf{x}_n|\omega_M)}{p(\mathbf{x}_n|\omega_M) + \sum_{l=1}^{M-1} P_l^*(n-1)\Big[p(\mathbf{x}_n|\omega_l) - p(\mathbf{x}_n|\omega_M)\Big]}, \quad i=1, \ldots, M-1, \tag{21}$$

and $\hat{\mathbf{P}}(n)$ is given by (7). The random vector $\mathbf{h}(\mathbf{X}, \mathbf{P}^*(n-1))$ is finite for all $\mathbf{X}$ and possesses a mean vector and a covariance matrix with finite components. Further, $\mathbf{H}(\mathbf{P}^*(n-1))$ is invertible (see Appendix I). These imply that $\hat{\mathbf{P}}(n)$ is finite for all finite n. For purposes of analysis, (7) is equivalent to

$$\hat{\mathbf{P}}(n) = \hat{\mathbf{P}}(n-1) - \frac{1}{n}[\hat{\mathbf{P}}(n-1) - \mathbf{P}] - \frac{1}{n}\mathbf{H}^{-1}(\mathbf{P}^*(n-1))\Big[\mathbf{h}(\mathbf{x}_n, \mathbf{P}^*(n-1)) - E[\mathbf{h}(\mathbf{X}, \mathbf{P}^*(n-1))]\Big]. \tag{22}$$

We show that $\hat{\mathbf{P}}(n)$ given by (22) converges to $\mathbf{P}$ in mean square and with probability one by a simple application of the multidimensional stochastic approximation theorem. We then derive an expression for the asymptotic variance of $\hat{\mathbf{P}}(n)$ using a theorem due to Sacks [6].

A. Convergence

Stochastic Approximation Theorem: Let $\mathbf{u}(n)$ be an n sample vector estimate with the corresponding sequence of random vectors governed by

$$\mathbf{U}(n) = \mathbf{t}_n[\mathbf{U}(0), \ldots, \mathbf{U}(n-1)] + \mathbf{R}_n, \quad n > 0 \tag{23}$$

where

(1) $\mathbf{U}(0)$ is an initial random vector, whose outcome is used to start the process of computing successive estimates,

(2) $\mathbf{t}_n[\mathbf{u}(0), \ldots, \mathbf{u}(n-1)]$ is a sequence of measurable transformations, and

(3) $\mathbf{R}_n$ is a function of the nth observation.

Let

$$E\left[|\mathbf{U}(0)|^2\right] < \infty \tag{24}$$

$$E[\mathbf{R}_n] = 0, \quad n > 0\,; \qquad \sum_{n=1}^{\infty} E\left[|\mathbf{R}_n|^2\right] < \infty. \tag{25}$$

Let α_n, β_n, and γ_n be sequences of nonnegative real numbers such that

$$\lim_{n\to\infty} \alpha_n = 0, \quad \lim_{n\to\infty} \sum_{i=1}^{n} \beta_i < \infty, \quad \text{and} \quad \lim_{n\to\infty} \sum_{i=1}^{n} \gamma_i = \infty. \tag{26}$$

Further, if the function $\mathbf{t}_n$ satisfies

$$|\mathbf{t}_n[\mathbf{u}(0), \ldots, \mathbf{u}(n-1)] - \mathbf{u}^*| \le \max\,\{\alpha_n,\, (1+\beta_n)|\mathbf{u}(n-1) - \mathbf{u}^*| - \gamma_n\} \tag{27}$$

for all real $\mathbf{u}(0), \ldots, \mathbf{u}(n-1)$, then the sequence of random vectors $\{\mathbf{U}(n)\}$ given by (23) converges to $\mathbf{u}^*$ in mean square, *i. e.*,

$$\lim_{n\to\infty} E\left[|\mathbf{U}(n) - \mathbf{u}^*|^2\right] = 0 \tag{28}$$

and with probability one, *i. e.*,

$$P[\mathbf{U}(n) \to \mathbf{u}^*] = 1 \quad \text{as} \quad n \to \infty. \tag{29}$$

The above statement of multidimensional stochastic approximation theorem is a slightly modified version of that in Sklansky and Wassel [7].

Convergence Theorem: The sequence of Prior Probabilities' estimates given by (19), (20), (21), (22) converges to the true vector of prior probabilities, $[P_1, \ldots, P_{M-1}]^T$.

Proof: We work out a correspondence from variables and sequences in (22) to those in the Stochastic Approximation theorem. Let $\hat{\mathbf{P}}(n) = \mathbf{U}(n)$ as well as $\mathbf{u}(n)$ depending on the context. $\hat{\mathbf{P}}(0)$ given by (19) satisfies (24). Let

$$\mathbf{t}_n[\mathbf{u}(0), \ldots, \mathbf{u}(n-1)] = \mathbf{t}_n[\hat{\mathbf{P}}(0), \ldots, \hat{\mathbf{P}}(n-1)] = \hat{\mathbf{P}}(n-1) - \frac{1}{n}[\hat{\mathbf{P}}(n-1) - \mathbf{P}] \quad \text{and} \tag{30}$$

$$\mathbf{R}_n = \frac{1}{n}\,\mathbf{H}^{-1}(\mathbf{P}^*(n-1))\left[\mathbf{h}(\mathbf{X}_n, \mathbf{P}^*(n-1)) - E[\mathbf{h}(\mathbf{X},\mathbf{P}^*(n-1))]\right] \tag{31}$$

so that our estimator is of the form (23). Taking expectations, $E[\mathbf{R}_n] = 0, \ n > 0$

$$E\left[|\mathbf{R}_n|^2\right] = \frac{1}{n^2} E\left\{\left[\mathbf{h}(\mathbf{X}_n,\mathbf{P}^*(n-1)) - E[\mathbf{h}(\mathbf{X},\mathbf{P}^*(n-1))]\right]^T\right.$$

$$\mathbf{H}^{-T}(\mathbf{P}^*(n-1))\mathbf{H}^{-1}(\mathbf{P}^*(n-1))\Big[\mathbf{h}(\mathbf{X}_n,\mathbf{P}^*(n-1)) - E[\mathbf{h}(\mathbf{X},\mathbf{P}^*(n-1))]\Big]\Big\} \quad (32)$$

where the expectation is with respect to the random variable $\mathbf{X}_n$.

$$E[|\mathbf{R}_n|^2] \le \frac{1}{n^2} B \quad (33)$$

where B is the supremum, w.r.t. $\mathbf{P}^*(n-1)$, of expectation on the RHS of (32). From Appendix I, $\hat{\mathbf{h}}(\mathbf{X}_n,\mathbf{P}^*(n-1))$ possesses finite moments and $\mathbf{H}(\mathbf{P}^*(n-1))$ is invertible. Therefore B is finite and

$$\sum_{n=1}^{\infty} E[|\mathbf{R}_n|^2] \le B \sum_{n=1}^{\infty} \frac{1}{n^2} < \infty \quad \text{verifying (25).}$$

$$|\mathbf{t}_n[\mathbf{u}(0), \ldots, \mathbf{u}(n-1)] - \mathbf{u}^*| = |\hat{\mathbf{P}}(n-1) - \frac{1}{n}[\hat{\mathbf{P}}(n-1) - \mathbf{P}] - \mathbf{P}| = \frac{n-1}{n}|\hat{\mathbf{P}}(n-1) - \mathbf{P}| \quad (34)$$

is known to be finite for all finite n. We need to determine α_n, β_n, and γ_n satisfying (26). Let

$$\alpha_n = \frac{1}{n}, \quad \beta_n = 0, \text{ and} \quad (35)$$

$$\gamma_n = \frac{1}{n}|\hat{\mathbf{P}}(n-1) - \mathbf{P}| \quad \text{if} \quad |\hat{\mathbf{P}}(n-1) - \mathbf{P}| \neq 0$$

$$= \frac{n-1}{n}\gamma_{n-1} \quad \text{if} \quad |\hat{\mathbf{P}}(n-1) - \mathbf{P}| = 0. \quad (36)$$

These definitions satisfy (26). (27) for the present case is

$$\frac{n-1}{n}|\hat{\mathbf{P}}(n-1) - \mathbf{P}| \le \max\{\frac{1}{n}, |\hat{\mathbf{P}}(n-1) - \mathbf{P}| - \gamma_n\}. \quad (37)$$

For any n, if $|\hat{\mathbf{P}}(n-1) - \mathbf{P}| = 0$, (37) is satisfied by 0 being bounded from above by $\alpha_n = 1/n$. If $|\hat{\mathbf{P}}(n-1) - \mathbf{P}| \neq 0$, then

$$|\hat{\mathbf{P}}(n-1) - \mathbf{P}| - \gamma_n = |\hat{\mathbf{P}}(n-1) - \mathbf{P}| - \frac{|\hat{\mathbf{P}}(n-1) - \mathbf{P}|}{n} = \frac{n-1}{n}|\hat{\mathbf{P}}(n-1) - \mathbf{P}|$$

satisfying (37). Thus, the equivalent condition (27) is satisfied for all possible sequences $\{\hat{\mathbf{P}}(n)\}$ generated by (22). As a consequence of the Stochastic Approximation theorem, (28) and (29) are true implying the convergence of $\{\hat{\mathbf{P}}(n)\}$ to $\mathbf{P}$ with probability one and in mean square.

B. Asymptotic Variance

In order to show that $\hat{\mathbf{P}}(n)$ given by (22) is asymptotically efficient, we establish

$$\lim_{n \to \infty} n\,\mathbf{Cov}[\hat{\mathbf{P}}(n)] = \lim_{n \to \infty} n\,\mathbf{J}^{-1} \quad (38)$$

where the limits are taken element by element within the matrix. We express (22) to reflect a multidimensional Robbins-Monro Stochastic Approximation form:

$$\hat{\mathbf{P}}(n) = \hat{\mathbf{P}}(n-1) - \frac{1}{n}\Big\{\hat{\mathbf{P}}(n-1) - \mathbf{P} - \mathbf{H}^{-1}(\mathbf{P}^*(n-1))\{\mathbf{h}(\mathbf{x}_n,\mathbf{P}^*(n-1)) - E[\mathbf{h}(\mathbf{X},\mathbf{P}^*(n-1))]\}\Big\} \quad (39)$$

Let a random vector

$$\mathbf{Z}(\hat{\mathbf{P}}(n-1)) = \mathbf{H}^{-1}(\mathbf{P}^*(n-1))\left[\mathbf{h}(\mathbf{X}_n,\mathbf{P}^*(n-1)) - E[\mathbf{h}(\mathbf{X},\mathbf{P}^*(n-1))]\right] \text{ and} \tag{40}$$

$$\mathbf{M}(\hat{\mathbf{P}}(n-1)) = \hat{\mathbf{P}}(n-1) - \mathbf{P}. \tag{41}$$

The problem of estimating $\mathbf{P}$ can now be reformulated as "find the root of the vector function $\mathbf{M}(\hat{\mathbf{P}}(n))$ by using noisy observations $\mathbf{M}(\hat{\mathbf{P}}(n-1)) - \mathbf{Z}(\hat{\mathbf{P}}(n-1))$ where $\mathbf{Z}(\hat{\mathbf{P}}(n-1))$ is a zero mean random vector with finite variance components." (39), expressed with these new variables manifests as

$$\hat{\mathbf{P}}(n) = \hat{\mathbf{P}}(n-1) - \frac{1}{n}\left[\mathbf{M}(\hat{\mathbf{P}}(n-1)) - \mathbf{z}(\hat{\mathbf{P}}(n-1))\right] \tag{42}$$

where the quantity within square brackets is a known function of the n-th observation $\mathbf{x}_n$. We now state a special case of a theorem due to Sacks [6] to facilitate computation of asymptotic covariance matrix of $\hat{\mathbf{P}}(n)$ in (38).

Sacks' theorem: Let

$$\mathbf{M}(\mathbf{v}) = \mathbf{v} - \boldsymbol{\theta}, \quad \sup_{\mathbf{v}} E\left[\|\mathbf{Z}(\mathbf{v})\|^2\right] < \infty, \quad \lim_{\mathbf{v}\to\boldsymbol{\theta}} E[\mathbf{Z}(\mathbf{v})\mathbf{Z}^T(\mathbf{v})] = \Pi, \tag{43}$$

(Π a positive definite matrix), and

$$\lim_{R\to\infty} \lim_{\epsilon\to 0^+} \sup_{|\mathbf{v}-\boldsymbol{\theta}|<\epsilon} \int_{|\mathbf{z}(\mathbf{v})|>R} |\mathbf{z}(\mathbf{v})|^2 \, dP = 0. \tag{44}$$

Then the sequence of random vectors, $\{\mathbf{V}(n)\}$ given by

$$\mathbf{V}(n) = \mathbf{V}(n-1) - \frac{1}{n}\left[\mathbf{M}(\mathbf{V}(n-1)) - \mathbf{Z}(\mathbf{V}(n-1))\right] \tag{45}$$

is such that $\sqrt{n}\,(\mathbf{V}(n) - \boldsymbol{\theta})$ is asymptotically normal with zero mean and covariance matrix Π.

Theorem: The estimator of prior probabilities given by (19), (20), (21), and (22) is asymptotically efficient.

Proof: We work out a correspondence between the quantities in (39) and those in Sacks' theorem. (41) corresponds to the first equation in (43). From Appendix I, the range of $\mathbf{P}^*(n-1)$ given by (20) restricts $\mathbf{H}(\mathbf{P}^*(n-1))$ to be invertible and $\mathbf{h}(\mathbf{X},\mathbf{P}^*(n-1))$ to possess a mean vector and covariance matrix with finite components. Thus,

$$\sup_{\mathbf{P}^*(n-1)} E\left[\|Z(\hat{\mathbf{P}}(n-1))\|^2\right] < \infty$$

verifying the second equation in (43). The LHS of (45) reduces to $E[\mathbf{Z}(\mathbf{P})\,\mathbf{Z}^T(\mathbf{P})]$ due to the continuity of $\mathbf{z}(\ )$ around $\mathbf{P}$.

$$E[\mathbf{Z}(\mathbf{P})\,\mathbf{Z}^T(\mathbf{P})] = \mathbf{H}^{-1}\,\mathbf{Cov}[\mathbf{h}(\mathbf{X},\mathbf{P})]\,\mathbf{H}^{-T} = n\mathbf{J}^{-1}\frac{1}{n}\,\mathbf{J}n\mathbf{J}^{-1} = n\mathbf{J}^{-1} \tag{46}$$

from (16) and (17). Therefore, Π in (43) corresponds to $n\mathbf{J}^{-1}$.

$$\lim_{R\to\infty} \lim_{\epsilon\to 0^+} \sup_{|\hat{\mathbf{P}}(n-1)-\mathbf{P}|<\epsilon} \int_{\{|\mathbf{z}(\hat{\mathbf{P}}(n-1))|>R\}} |\mathbf{z}(\hat{\mathbf{P}}(n-1))|^2 dP = \lim_{R\to\infty} \int_{\{|\mathbf{z}(\mathbf{x},\mathbf{P})|>R\}} |\mathbf{z}(\mathbf{x},\mathbf{P})|^2 p(\mathbf{x})d\mathbf{x}$$

$$= \lim_{R\to\infty} \int_{\{|\mathbf{x}|>R\}} \left|\mathbf{H}^{-1}\left[\mathbf{h}(\mathbf{x},\mathbf{P}) - E[\mathbf{h}(\mathbf{X},\mathbf{P})]\right]\mathbf{H}^{-1}\right|^2 p(\mathbf{x})d\mathbf{x} = 0 \tag{47}$$

due to the finiteness of $\mathbf{h}(\mathbf{x},\mathbf{P})$, and of the class conditional mean vectors. Therefore, from Sacks' theorem, $\{\hat{\mathbf{P}}(n)\}$ given by (39) is such that $\sqrt{n}\,(\hat{\mathbf{P}}(n)-\mathbf{P})$ is asymptotically normal with zero mean and covariance matrix $n\mathbf{J}^{-1}$. Thus,

$$\lim_{n\to\infty} \mathbf{Cov}[\sqrt{n}\,\hat{\mathbf{P}}(n)] = \lim_{n\to\infty} [n\mathbf{J}^{-1}] \text{ or } \lim_{n\to\infty} n\,\mathbf{Cov}[\hat{\mathbf{P}}(n)] = \lim_{n\to\infty} n[\mathbf{J}^{-1}] \tag{48}$$

implying asymptotic efficiency of the estimator (39) or equivalently, that of (22).

V. SIMULATION EXPERIMENT AND DISCUSSION

Among the many simulation experiments conducted, the following three class, three feature pattern recognition problem is chosen to present illustrative results. The prior probabilities are

$$P(\omega_1) = 0.2, \quad P(\omega_2) = 0.5, \quad \text{and} \quad P(\omega_3) = 0.3$$

Features are binary (zero-one) and class conditionally independent. Their distributions are given in the following table.

Feature	$P(X_i{=}1\vert\omega_j)$		
	$j{=}1$	$j{=}2$	$j{=}3$
X_1	0.2	0.6	0.8
X_2	0.8	0.6	0.2
X_3	0.6	0.8	0.2

The probability of error of the Bayes classifier for this problem is 0.6816. Five PPEs are chosen for a comparative study of the performance of the asymptotically efficient and other schemes. They are

(1) *The supervised estimator.* This is the unbiased and efficient estimator of prior probabilities when the class label of each observed sample is made available. This estimator is not applicable in practice when only unsupervised data is available, as in the present study. Its results are presented only for comparison.

(2) *Linear Estimator.* The chosen class conditional distributions of the features are such that prior probabilities can be estimated [2] by a linear function of observations of unsupervised data. The linear estimator is very simple to implement.

(3) *Nonadaptive Estimator.* This is similar to the estimator given by (19), (20), (21), and (22) except that the $\mathbf{P}^*$ is a constant vector with each component given by $1/M$. The advantage of this scheme is that the $\mathbf{H}^{-1}$ matrix in (22) will be constant, simplifying the computation required during each iteration. Computation of the vector

$$\mathbf{h}(\mathbf{x}_n,\mathbf{P}^*(n-1)) - E[\mathbf{h}(\mathbf{X},\mathbf{P}^*(n-1))]$$

can be similarly simplified for discrete feature problems, by adopting a table look up procedure.

(4) *Fictitious Efficient Estimator.* This scheme is obtained by substituting $\mathbf{P}^*$ with the actual prior probabilities being estimated. The scheme, therefore, can not be implemented in practice.

(5) *Asymptotically Efficient Estimator.* This is the subject of the present study.

The supervised estimator starts the estimation process without any initial values. The other schemes use $1/M$ for each estimate to initiate recursion. However, this does not significantly influence the later estimates; the estimates are erratic for the first few tens of samples. Results are plotted from the 100th sample. The problem statistics are such that the supervised estimator itself takes over a thousand samples before yielding estimates close to true values. The relatively high probability of error of the Bayes classifier indicates that the performance of the best of the unsupervised estimators should be considerably worse than that of the supervised estimator. The linear estimator is the simplest but the least efficient of those studied here and is quite slow in convergence. The fictitious efficient estimator, the asymptotically efficient estimator, and the nonadaptive estimator are quite close to each other (though not identical) in the simulation results. Their performances are not far from that of the supervised estimator. The answer to the important question of whether or not the extra programming and/or the computation required by the asymptotically efficient scheme is worth the sacrifice depends on the type of data and the environment of the decision scheme. If the unknown prior probabilities are known not to widely differ from one another, the nonadaptive estimator (with $P_i^* = 1/M$, $i=1, \ldots, M-1$) is a very good choice due to its low computation and good performance. It is not advisable to use the asymptotically efficient scheme for problems involving continuous features. Digital Communication systems process a large number of samples with relatively stationary parameters. With the increasing use of computers at receivers, the asymptotically efficient estimator should find use in such systems. Expert diagnostic systems is another area in which decision systems are often required to collate a large amount of existing data to be able to make better decisions. The computation involved in the asymptotically efficient estimator is quite minimal in comparison with the searching and sorting which the Expert Systems' programs are required to do routinely. The complexity of many problems formulated as Pattern Recognition within large Expert Systems is of the order of the one used here for simulation. Asymptotically efficient PPEs are therefore useful in some expert diagnostic systems.

VI. CONCLUDING REMARKS

The main contribution of this paper is the derivation of a recursive estimator of prior probabilities in a multiclass finite mixture, the proof that it is asymptotically efficient, and simulation results. Previous work reported in this area has merely conjectured on the existence of asymptotically efficient estimators for the present problem. Our approach depends on [2] which proposes a class of estimators based on functions of class conditional density functions of observations. By a logically derived choice of such a function we worked out an estimator which was a strong candidate for asymptotic efficiency. We used the Sacks' theorem to prove the asymptotic efficiency of the estimator. Sacks states that the conditions required for his theorem imply convergence with probability one (and of course in mean square) as a consequence of Dvoretzky's theorem on multidimensional Stochastic Approximation. Our explicit proof of convergence by a direct application of a form of Dvoretzky's theorem helps in better understanding of the alternative forms of the recursive estimator.

APPENDIX I

Properties of $\mathbf{h}(\mathbf{X},\mathbf{P}^*)$. Proofs are omitted to conserve space.

$$h_i(\mathbf{X},\mathbf{P}^*) = \frac{p(\mathbf{X}|\omega_i) - p(\mathbf{X}|\omega_M)}{p(\mathbf{X}|\omega_M) + \sum_{l=1}^{M-1} P_l^* [p(\mathbf{X}|\omega_l) - p(\mathbf{X}|\omega_M)]}, \quad i = 1, \ldots, M-1 \tag{A1}$$

with

$$0 < P_l^* < 1, \quad l = 1, \ldots, M-1 \text{ and } \sum_{i=1}^{M-1} P_l^* < 1. \tag{A2}$$

$p(\mathbf{x}|\omega_i)$, $i=1, \ldots, M$ are linearly independent density functions possibly possessing Dirac impulse function components. Let

$$p(\mathbf{x}|\omega_i) = f_i(\mathbf{x}) + \sum_{k=1}^{K} \delta(\mathbf{x}-\mathbf{u}_k)a_{ik}\,, \quad i=1,\ldots,M. \tag{A3}$$

$\delta(\;)$ is the Dirac impulse function. $f_i(\mathbf{x})$ are devoid of Dirac impulse functions.

Propositions:

(1) The denominator on the RHS of (A1) is a valid mixture density function. Further, since the mixing quantities P_l^* are all nonzero, the denominator going to zero for certain $\mathbf{x}$ implies that the numerator goes to zero as well.

(2) With the density functions as defined in (A3),

$$\begin{aligned} h_i(\mathbf{x}|\omega_i) &= 0 \text{ if } p(\mathbf{x}|\omega_i) = p(\mathbf{x}|\omega_M) = 0 \\ &= \frac{p(\mathbf{x}|\omega_i) - p(\mathbf{x}|\omega_M)}{p(\mathbf{x}|\omega_M) + \sum_{l=1}^{M-1} P_l^*[p(\mathbf{x}|\omega_l) - p(\mathbf{x}|\omega_M)]}\,, \text{ if } \mathbf{x}\neq\mathbf{u}_k,\ k=1,\ldots,K. \\ &= \frac{a_{ik} - a_{Mk}}{a_{Mk} + \sum_{l=1}^{M-1} P_l^*[a_{ik} - a_{Mk}]}\,, \text{ if } \mathbf{x}=\mathbf{u}_k. \end{aligned} \tag{A4}$$

Thus $h_i(\mathbf{x}|\omega_i)$ is finite and can be evaluated for all $\mathbf{x}$. It is finitely discontinuous at K points.

(3) Moments of all finite orders of $h_i(\mathbf{X},\mathbf{P}^*)$ are finite. This follows from the finiteness of $h_i(\mathbf{x},\mathbf{P}^*)$ for all $\mathbf{x}$.

(4) The matrix $\mathbf{H}(\mathbf{P}^*)$ is invertible.

(5) $\mathbf{J} = n\mathbf{H}(\mathbf{P})$ and hence $\mathbf{J}$ is invertible.

(6) $E[\mathbf{h}(\mathbf{X},\mathbf{P})] = 0$ and $\mathbf{Cov}[\mathbf{h}(\mathbf{X},\mathbf{P})]$ is invertible.

REFERENCES

[1] D. C. Boes, "On the estimation of mixing distributions," *Ann. Math. Statist.*, vol. 37, pp. 177-188, Jan. 1966.

[2] G. R. Dattatreya and L. N. Kanal, "Estimation of prior probabilities in multicategory information processing," Computer Science Report, University of Texas at Dallas, 1986.

[3] D. Kazakos, "Recursive estimation of prior probabilities using a mixture," *IEEE Trans. Information Theory,* vol. IT-23, pp. 203-211, Mar. 1977.

[4] D. Kazakos and L. D. Davisson, "An improved decision-directed detector," *IEEE Trans. Information Theory,* vol. IT-26, pp. 113-116, Jan. 1980.

[5] U. E. Makov, "On the choice of gain functions in recursive estimation of prior probabilities," *IEEE Trans. Information Theory,* vol. IT-26, pp. 497-498, July 1980.

[6] J. Sacks, "Asymptotic distribution of stochastic approximation procedures," *Ann. Math. Statist.*, vol. 29, pp. 373-405, June 1958.

[7] J. Sklansky and G. N. Wassel, *Pattern Classifiers and Trainable Machines.* Berlin: Springer Verlag, 1981.

[8] H. L. Van Trees, *Detection, Estimation, and Modulation Theory, Part 1.* NY: John Wiley, 1968.

Robust Locally-Optimal Detection of Dependent Signals in Non-Gaussian Noise *

P. Willett and P. Srivastava, Univ. of Connecticut, Storrs CT 06268

Sept 1988

Let us define an asymptotic sequence of detection problems as

$$\begin{aligned} H_n &: X_i = N_i \\ K_n &: X_i = N_i + \theta_n S_i \qquad i = 1, 2, \ldots, n \end{aligned}$$

where $\{X_i\}_{i=1}^n$ is the observations process, $\{N_i\}_{i=1}^n$ and $\{S_i\}_{i=1}^n$ are the (stochastic) noise and signal processes respectively, $\theta_n = \gamma n^{-1/4}$, and γ is some constant. Given that the extended regularity conditions given in [1] are satisfied, then the performance measure is the *efficacy*, defined in this case as

$$\eta = \lim_{n \to \infty} \frac{1}{4n} \left[\frac{\ddot{\mu}_n(0)}{\sigma_n(0)} \right]^2$$

where

$$\ddot{\mu}_n(\theta) = \frac{\partial^2}{\partial \theta^2} \left[\mathcal{E}\{\lambda_n(\vec{\mathbf{x}}|\theta)\} \right] \quad \text{and} \quad \sigma_n^2(\theta) = \mathcal{V}\{\lambda_n(\vec{\mathbf{x}}|\theta)\}$$

Here $\mathcal{E}$ and $\mathcal{V}$ denote expectation and variance, and $\lambda_n(\vec{\mathbf{x}})$ is the test statistic.

In practice it would not be safe to assume that the signal is white; in the passive sonar case, for example, the stochastic signal could contain narrowband components corresponding to an enemy vessel's drive mechanism. Kassam in [2] has used the model

1. $\{N_i\}_{i=1}^n$ is i.i.d. with univariate density $f(\cdot)$
2. $\{S_i\}_{i=1}^n$ has an arbitrary density with $\mathcal{E}(S_i) = 0$, $\mathcal{V}(S_i) = 1$, and $\mathcal{E}(S_i S_{i+k}) = r(k)$

Note that due to the multiplication by θ, the unit-variance signal condition is not restrictive.

The locally-optimal test statistic for such a model has been shown to be [2]

$$\lambda(\vec{\mathbf{x}}) = \sum_{i=1}^{n} h_{lo}(X_i) + \sum_{i=1}^{n} \sum_{j=1}^{n} r(i-j) g_{lo}(X_i) g_{lo}(X_j)$$

where

$$h_{lo}(x) = \frac{\ddot{f}(x)}{f(x)} - \left[\frac{\dot{f}(x)}{f(x)} \right]^2 \quad \text{and} \quad g_{lo}(x) = -\frac{\dot{f}(x)}{f(x)}$$

Note that in the white signal case this reduces to the well-known result in [1].

Computing the performance of this test is not trivial, and previous treatments have dealt only with the white signal case. It is possible, however, to obtain:

$$\eta = \frac{1}{4} \left[\int \frac{(\ddot{f}(x))^2}{f(x)} dx + 2 \sum_{k=1}^{\infty} r(k)^2 I(f)^2 \right]$$

where I denotes Fisher's information. Note that with a white signal the second term vanishes.

Now let us consider the practical application of this result. In a stationary noise situation the detector will have an estimate of the noise density $f(\cdot)$; however, often the correlation structure of the signal $r(k)$ will not be known accurately. We pose the problem of finding a *min-max* detector, one which is optimal for the least favorable signal correlation possible. We call such a detector *robust* and claim that it will in general perform reasonably well over a wide class of correlations.

---

*Supported by the National Science Foundation under Grant ECS-8719031, and by the University of Connecticut Research Foundation under Grant 1171-000-22-0505-35-398

Clearly, if nothing whatever is known about $r(k)$, the robust detector is that corresponding to $(r(k) \equiv 0,\ k \neq 0)$. This is a somewhat trivial result – the worst case signal is white. It is reasonable to assume, however, that *some* knowledge is available about signal correlation, and proceeding in a manner similar to [4,5] we define a class Φ of allowable signal spectra $\phi(\omega)$ as

$$\Phi = \left\{ \phi(\omega) : [\phi(\omega) \leq \phi_U(\omega)\ \ (\forall \omega)] \ \ \cap \ \ [\phi(\omega) \geq \phi_L(\omega)\ \ (\forall \omega)] \ \ \cap \ \ \left[\frac{1}{2\pi} \int_{-\pi}^{\pi} \phi(\omega) d\omega = 1 \right] \right\}$$

Note that we must specify

$$\left[\frac{1}{2\pi} \int_{-\pi}^{\pi} \phi_U(\omega) d\omega \geq 1 \right] \qquad \left[\frac{1}{2\pi} \int_{-\pi}^{\pi} \phi_L(\omega) d\omega \leq 1 \right] \qquad [\phi_U(\omega) \geq \phi_L(\omega) \geq 0]$$

for the set Φ to be non-empty. Since we can write

$$2 \sum_{k=1}^{\infty} r(k)^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} \phi^2(\omega) d\omega - 1$$

we can express the spectrum corresponding to the robust detector as

$$\phi^*(\omega) = \arg\min_{\phi \epsilon \Phi} \left\{ \int_{-\pi}^{\pi} \phi^2(\omega) d\omega \right\} = \left\{ \begin{array}{cc} \phi_U(\omega) & \phi_U(\omega) \leq \lambda \\ \lambda & \phi_L(\omega) < \lambda < \phi_U(\omega) \\ \phi_L(\omega) & \phi_L(\omega) \geq \lambda \end{array} \right\}$$

The constant λ is chosen to satisfy the unit-variance constraint on $\phi(\omega)$. We note that this solution is identical to that derived in [3] for the robust Weiner filter in Gaussian noise; it is also similar to that in [4] for the Eckart filter.

In the case that more than one sensor is present, the detection problem can be re-posed as

$$\begin{array}{ll} H_n\!: X_{ji} = N_{ji} & \\ K_n\!: X_{ji} = N_{ji} + \theta_n S_i & i = 1, 2, \ldots, n \quad j = 1, 2, \ldots, L \end{array}$$

where X_{ji} is the i^{th} observation from the j^{th} sensor. This model indicates that the same signal is present at each element in the array; while this would seem appropriate only for a target broadside to the array, we can assume that time delays are being used to "steer" the array in a particular direction. We further assume that *all* the noise samples $\{\{N_{ji}\}_{j=1}^{L}\}_{i=1}^{n}$ are i.i.d.

The locally-optimal detector for this case is:

$$\lambda(\vec{\mathbf{x}}) = \sum_{i=1}^{n} \sum_{j=1}^{L} h_{lo}(X_{ji}) + \sum_{i=1}^{n} \sum_{k=1}^{n} r(i-k) \left[\sum_{j=1}^{L} g_{lo}(X_{ji}) \right] \left[\sum_{l=1}^{L} g_{lo}(X_{lk}) \right]$$

where $h_{lo}(\cdot)$ and $g_{lo}(\cdot)$ are as previously defined.

It turns out that the efficacy of this locally-optimal detector is very similar to that in the single-sensor case:

$$\eta = \frac{L}{4} \left[\int \frac{(\ddot{f}(x))^2}{f(x)} dx + \left([L-1] + 2L \sum_{k=1}^{\infty} r(k)^2 \right) I(f)^2 \right]$$

Again the efficacy is monotonically increasing in the integral of the square of the signal spectrum, so the robustness results for the single sensor apply also to the array detection case.

References

[1] S. Kassam, *Signal Detection in Non-Gaussian Noise*, Springer-Verlag, 1987.

[2] S. Kassam, "Locally-Robust Array Detectors for Random Signals", *IEEE Trans on IT*, Vol 24, 1978.

[3] S. Kassam and T. Lim, "Robust Weiner Filters", *Journal of the Frankin Institute*, Oct.-Nov. 1977.

[4] E. Al-Husseini and S. Kassam, "Robust Eckart Filters for Time Delay Estimation", *IEEE Trans. on ASSP*, Oct 1984.

[5] H. Poor, "Robust Matched Filters", *IEEE Trans on IT*, Sept 1983.

CRAMER-RAO BOUNDS FOR CONSTRAINED SPECTRUM ESTIMATION WITH APPLICATION TO A PROBLEM IN RADAR IMAGING †

JOSEPH A. O'SULLIVAN
PIERRE MOULIN
DONALD L. SNYDER

Electronic Systems and Signals Research Laboratory
Department of Electrical Engineering
Washington University
St. Louis, MO 63130

ABSTRACT

Some recent theoretical results for maximum likelihood estimation problems are discussed. These results apply to a variety of covariance estimation problems where the covariance is constrained to be the sum of nonnegative parameters times fixed hermitian symmetric matrices. Cramer-Rao bounds for estimating the parameters are obtained. Uniqueness of parameter estimates is related to these bounds. Also, these bounds motivate an approach to optimal signal design for radar imaging systems.

1. INTRODUCTION

The motivation for the present study is the examination of several issues which arose in connection with the estimation of spectrum samples for a problem in radar imaging [1]. This problem was originally thought to be unrelated to the spectrum estimation work taking place in our laboratory [2,3], but has since moved closer to that work. The recent paper [4] discusses some of the connections between the fundamental spectrum estimation problem discussed in [2] and the radar imaging problem [1]. Burg, Luenberger, and Wenger [5] introduced the problem of structured covariance estimation in the spirit of the present work. They derived necessary conditions for maximum likelihood estimates of the structured covariance matrices and proposed an algorithm for finding the estimates. One restrictive assumption which they made was that the sum of outer products of the data formed a full rank matrix. Fuhrmann and Miller [3] and Miller and Snyder [2] removed that restriction and explored estimating covariance matrices given only one data vector. In addition, Fuhrmann and Miller [3] derived conditions for the log-likelihood to be bounded above over a set of parameterized covariance matrices with probability one. Proving boundedness of the log-likelihood is a crucial step when attempting to prove convergence of estimation algorithms. Another crucial point to consider when examining convergence is the characterization of the set of maximum likelihood estimates. In particular, is there a unique estimate? One of the major results of this paper is the characterization of nonuniqueness of maximum likelihood estimates in terms of Cramer-Rao bounds. It is shown that there does not exist a positive definite maximum likelihood estimate for parameters of interest if the Fisher information matrix (inverse of Cramer-Rao lower bound) is singular.

The practical problem which has motivated much of our study is a delay-Doppler radar imaging problem where the spectrum to be estimated corresponds to an image of the target. For this problem it is the scattering function, which is a delay dependent power spectrum, that is of interest. For related problems, the parameters of interest are the lag covariances. These problems are not always equivalent and one goal of this paper is to point out some of the issues which are unique to the spectrum estimation formulation of the problem.

When studying the potential implementation of a radar imaging system, one issue which must be resolved is the selection of the transmitted signal. The problem of signal selection is analogous to the selection of an input for the identification of a linear system. Several authors have studied this problem from the point of view of maximizing the achievable performance. This has been interpreted to mean either maximizing a norm on the Fisher information matrix [6,7,8] or minimizing a norm on the Cramer-Rao lower bound [8]. A possible application of these ideas is presented.

---

† This work was supported by contract number N00014-86-K-0370 from the Office of Naval Research.

The results are presented in a rather general manner and are then specialized to problems of interest. First, problems of interest are discussed. This section also introduces some of the notation used throughout the remainder of the paper. The third section derives the Cramer-Rao lower bound for the variance of unbiased estimates of the spectrum samples. This bound is used in each of the other sections of the paper. The equivalence of nonuniqueness of maximum likelihood estimates for the spectrum samples and infinite magnitude of the Cramer-Rao lower bound is then demonstrated. The fourth section addresses a problem of signal design for radar imaging systems. The task is to choose a transmitted signal such that the Cramer-Rao lower bound for the variance of the spectrum estimates is minimized with respect to the transmitted signal. The final section discusses the implications of these results and suggests further work.

2. PROBLEM DEFINITION

There are three problems which are of interest. The first two are special cases of the third, but some of the results to be presented are applicable to only one of the special cases.

Let W_P be a P×P DFT matrix. The k,m entry of W_P is $(1/\sqrt{P}\,)e^{-j2\pi km/P}$. In each of the three problems, r is a linear combination of a signal vector and additive white Gaussian noise. The goal of each problem is to find the maximum likelihood estimate for the power spectrum of the desired signal given one observation vector r.

Problem one:

$$r = b + w, \tag{2.1}$$

where b is an N-vector sample from a zero mean Gaussian process with Toeplitz covariance matrix K_1, and w is white Gaussian noise with covariance $N_0 I_N$. The matrix K_1 is constrained to be of the form

$$K_1 = [\,I_N\ 0\,]\, W_P^\dagger \Sigma W_P \begin{bmatrix} I_N \\ 0 \end{bmatrix}, \tag{2.2}$$

where Σ is a P×P real diagonal matrix with nonnegative entries, I_N is an N×N identity matrix, and $\dagger$ denotes complex conjugate transpose. This is equivalent to assuming that b is $\Gamma_1^\dagger$ times c, where c is a zero mean Gaussian vector with covariance Σ and

$$\Gamma_1 = W_P \begin{bmatrix} I_N \\ 0 \end{bmatrix}. \tag{2.3}$$

Problem two:

$$r = S^\dagger b + w, \tag{2.4}$$

where w is as before, b is an M length vector,

$$b = [I_M\ 0] W_P^\dagger c, \tag{2.5}$$

but now there is an N×M matrix $S^\dagger$ multiplying b. The vector c is zero mean Gaussian distributed with real diagonal covariance given by Σ. Under these assumptions, the covariance of r is $K_2 + N_0 I_N$, where

$$K_2 = S^\dagger [I_M\ 0] W_P^\dagger \Sigma W_P \begin{bmatrix} I_M \\ 0 \end{bmatrix} S = \Gamma_2^\dagger \Sigma \Gamma_2, \tag{2.6}$$

where

$$\Gamma_2 = W_P \begin{bmatrix} I_M \\ 0 \end{bmatrix} S. \tag{2.7}$$

This problem reduces to the first when N = M and $S = I_N$.

Problem three:

$$r = S^\dagger b + w, \tag{2.8}$$

where w is as before, $S^\dagger$ is an N by MI_R signal matrix, and now b is a sample from a process whose covariance matrix is $MI_R \times MI_R$ block diagonal with each $M \times M$ block being Toeplitz. Separate b into I_R subvectors each of length M and label these subvectors sequentially from 0 to $I_R - 1$. The covariance of

$b(k)$ is given by

$$K_B(k) = [I_M\ 0]W_L^\dagger \Sigma(k) W_L \begin{bmatrix} I_M \\ 0 \end{bmatrix} \tag{2.9}$$

where $K_B(k)$ is the k^{th} diagonal block of the covariance of b. If the $L \times L$ matrix $\Sigma(k)$ is placed in the k^{th} diagonal block of a $LI_R \times LI_R$ block diagonal matrix, Σ, then we can write

$$K_3 = \Gamma_3^\dagger \Sigma \Gamma_3, \tag{2.10}$$

where the matrix Γ_3 is defined as the product of a block diagonal matrix and S; the block diagonal matrix has I_R blocks each of which equals $W_L[I_M\ 0]^\dagger$. For this problem, let $P = LI_R$. This problem reduces to the second when there is only one block on the diagonal ($I_R = 1$). Since each of the other problems is obtained by simplifying this problem, we can focus our attention on this problem.

We follow the analysis in [1,2]. Let k denote either 1, 2, or 3. Problem k may be restated as follows. Assume the observation vector r is given,

$$r = \Gamma_k^\dagger c + w, \tag{2.11}$$

where c is a Gaussian random variable with covariance Σ, and w is white Gaussian noise with intensity $N_0 I_N$. Find the maximum likelihood estimate for Σ. Under the assumptions, r is a Gaussian distributed random variable with covariance

$$K_r = \Gamma_k^\dagger \Sigma \Gamma_k + N_0 I_N. \tag{2.12}$$

The loglikelihood function is (neglecting terms not involving Σ)

$$L(r,\Sigma) = -\ln\det(\Gamma_k^\dagger \Sigma \Gamma_k + N_0 I_N) - r^\dagger (\Gamma_k^\dagger \Sigma \Gamma_k + N_0 I_N)^{-1} r. \tag{2.13}$$

Taking the partial derivative with respect to Σ and setting it to zero yields the trace condition which a solution must satisfy [1,2]. A necessary condition for Σ to be the maximum likelihood estimate for the spectrum for problem k is that

$$Tr\,[\Gamma_k (\Gamma_k^\dagger \Sigma \Gamma_k + N_0 I)^{-1} (r r^\dagger - \Gamma_k^\dagger \Sigma \Gamma_k - N_0 I)(\Gamma_k^\dagger \Sigma \Gamma_k + N_0 I)^{-1} \Gamma_k^\dagger \delta\Sigma] = 0 \tag{2.14}$$

for all diagonal matrices $\delta\Sigma$ of the same size as Σ. Typically this equation cannot be solved for Σ explicitly and some iterative technique for determining the maximum likelihood estimate must be used. The algorithm used in [1,2] is the EM algorithm.

3. CRAMER-RAO LOWER BOUND

This section contains a derivation of the Cramer-Rao lower bound for unbiased estimates of the covariance matrices for the problems from Section 2. It is important to remember that, for biased estimates, the lower bound for the variance of the estimates is proportional to this bound, so these calculations are relevant for that case as well. In fact, the maximum likelihood estimation procedure currently implemented in our laboratory generates biased estimates for the spectrum samples under certain conditions [9].

Following Van Trees [10, p. 79], the variance of any unbiased maximum likelihood estimate of a nonrandom parameter is greater than or equal to the diagonal elements of the inverse of the Fisher information matrix. The Fisher information matrix has entries equal to the negatives of the expected value of second the derivatives of $L(r,\Sigma)$ with respect to the parameters (in this case Σ).

Lemma 1: For problem k, the m,n element of the Fisher informaion matrix equals the magnitude squared of m,n element of the matrix

$$\Gamma_k (\Gamma_k^\dagger \Sigma \Gamma_k + N_0 I_N)^{-1} \Gamma_k^\dagger. \tag{3.1}$$

Proof: In the appendix of [1] an expression for the second derivative of $L(r,\Sigma)$ is obtained. Taking the negative of the expected value yields the above expression.□

The above result is very important for the simulations in order to determine the performance of the estimation scheme used. Some of the simulations performed in our Laboratory are reported in [9] and the performance is compared to this bound. In addition, the algorithm proposed in [1] is compared to other algorithms for estimating spectrum samples and it is shown that in certain cases this algorithm outperforms the others.

One of the goals of calculating the Cramer-Rao bound is to use it to determine the well posedness of estimation problems.

Lemma 2: For problem one the rank of the Fisher information matrix is less than or equal to 2N-1. Furthermore, if $P > 2N-1$ then there is no unique Σ positive definite which maximizes the loglikelihood.

Proof: The proof involves writing the Fisher information matrix as the product of three matrices one of which is 2N-1×2N-1. The rank of this matrix is thus shown to be less than or equal to 2N-1. The third matrix is the Hermitian transpose of the first. The first matrix is P by 2N-1 and has k,m element given by

$$\frac{1}{P} e^{-j2\pi k(m-N+1)/P}, \tag{3.2}$$

where k denotes the row and ranges from 0 to P-1 and m ranges from 0 to 2N-2. Let the α,β element of $(\Gamma_1^\dagger \Sigma \Gamma_1 + N_0 I_N)^{-1}$ be denoted by $f_{\alpha,\beta}$. Then the m,n element of the second matrix is

$$\sum_{\alpha=max(0,N-1-m)}^{min(N-1,2N-2-m)} \sum_{\beta=max(0,N-1-n)}^{\min(N-1,2N-2-n)} f_{m+\alpha-N+1,n+\beta-N+1} f^*_{\alpha,\beta}, \tag{3.3}$$

where m and n range from 0 to 2N-2 and * denotes complex conjugate. Multiplying these three matrices completes the first part of the proof. For the second part of the proof, it is easily shown that the addition of a real diagonal matrix with entries all of the same magnitude with alternating signs to any matrix does not change the value of the loglikelihood if P>N-1. Thus no positive definite Σ yields a unique value for the loglikelihood. By the same arguments, no Σ with one diagonal entry equal to 0 and the rest positive yields a unique value of $L(r,\Sigma)$. If two elements are zero, however, the argument fails and other tests must be used.□

Lemma 2 gives a precise statement about the uniqueness of maximum likelihood estimates for the problems under consideration. In particular, it states precisely the number of parameters which can be estimated uniquely for a problem of type one. We believe the bound is tight. That is, for a problem of type one, we believe that a unique maximum likelihood estimate exists whenever $P \leq 2N-1$ and that the Fisher information matrix is of rank P for this case. The basis for this belief is that the first matrix computed in the proof has full row rank for $P \leq 2N-1$. For cases of interest, the second matrix constructed is of full rank. If this second matrix is of full rank, the inequality in the first sentence of the lemma statement can be changed to equality.

The proof of the last lemma did not require that the elements of the Γ matrix be entries from W_P. It only required a certain stationarity property.

Lemma 3: Let the i,m element of Γ_k be $g_{i,m}$ and suppose that $g_{i,m} g^*_{i,n} = h_{i,m-n}$ (the product depends only on the row number and the difference m-n). Then the rank of the Fisher information matrix is at most 2N-1 and if $P > 2N-1$ there is no unique Σ positive definite which maximizes the loglikelihood.

Proof: Modify the proof of the last lemma by changing the first matrix to have k,m element $h_{k,m-N+1}$. The proof of the first part is then identical to that proof. To show that there is no unique Σ, we construct a matrix such that if this matrix is added to Σ then the value of the loglikelihood does not change. This diagonal perturbational matrix is formed as follows. Let γ_i denote the i^{th} row of Γ_k. Then we have

$$\Gamma_k^\dagger \Sigma \Gamma_k = \sum_{i=0}^{P-1} (\Sigma)_{i,i} \gamma_i^\dagger \gamma_i . \tag{3.4}$$

The potential elements of this matrix are completely determined by the elements of the outer products of γ_i with itself for i from 0 to P-1. Form P row vectors with the elements of these outer products by taking the Kronecker products of γ_i with γ_i^*. Take these P vectors, transpose them, and put them in the columns of a $N^2 \times P$ matrix. Denote this matrix G^T. This matrix has rank at most 2N-1 over the complex numbers due to the stationarity assumption on elements of Γ_k stated in the lemma. The 2N-1 linearly independent rows have i^{th} element $h_{i,m-N+1}$. One of these linearly independent rows has all real elements while the others are in general complex and appear in complex conjugate pairs (the real row has entries $h_{i,0} = |g_{i,m}|^2$). Thus this matrix has column rank at most 2N-1 when taken over the real numbers. Any real vector in the kernel of this matrix may be placed along the diagonal of a matrix and then added to Σ without changing the value of $L(r,\Sigma)$ because the diagonal matrices formed in this way have the property that they are mapped to the zero matrix when premultiplied by $\Gamma_k^\dagger$ and postmultiplied by Γ_k.□

Again the bound is tight because in general the matrix formed of Kronecker products of the rows of Γ_k will have rank 2N-1 over the reals. It is recognized that Lemma 3 is not really different from Lemma 2 in the sense that the assumption on Γ_k implies that each row consists of complex exponentials evenly spaced over an arc of the unit circle scaled by a complex constant. There is no relationship from row to row, however. One benefit of the proof of Lemma 3 is the introduction of the matrix G. Using G, a much more general result may be obtained.

Lemma 4: Let G be the $P \times N^2$ matrix whose i^{th} row is the Kronecker product of γ_i with γ_i^*. Assume that K_r is positive definite. Then the Fisher information matrix is singular if and only if the rank of G is less than P. If a positive definite Σ yields a unique maximum of the loglikelihood function then G has rank P.

Proof: The Fisher information matrix may be written as

$$G(K_r^{-1} \otimes K_r^{-1*})G^\dagger. \tag{3.5}$$

If K_r is nonsingular then the rank of this matrix equals the rank of G and it is singular if and only if the rank of G is less than P. If the rank of G is less than P then pick a real vector s such that $s^T G = 0$ (such an s exists because the Fisher information matrix has real entries). If these vector elements are placed along the diagonal of a matrix and added to Σ the matrix K_r remains unchanged and thus the loglikelihood is unchanged. On the other hand, if the rank of G equals P, then there is no such vector s and any change of the diagonal entries of Σ yields a different K_r.□

4. SIGNAL SELECTION

This section presents a new approach to the problem of signal selection for radar imaging systems. This approach is somewhat general and it can be changed to fit several problems of interest.

Most approaches to radar imaging are based on deterministic techniques. Typically, the reflections from the target are assumed to be deterministic and the radar returns are processed using a two dimensional transform technique. The transform used (or in general the linear processing scheme used) may be modified for individual problems, but the overall processing scheme is fixed. The signal selection problem may be thought of in a couple of different ways. One way is to select the signal which optimizes the radar return for the processing scheme used. For example, stepped frequency waveforms described in [11] may be thought of as optimizing the signal assuming that two dimensional Fourier transforms are used to process the radar return. In other words, this transmitted signal attempts to make the return signals look like the two dimensional Fourier transform of the target image. A second type of signal selection is more general. In this case, the signal is selected to achieve pulse compression so that by combining the returned signal properly the result of processing achieves desired range and crossrange resolution [11].

Our approach for radar imaging differs from conventional approaches by starting with a model which accurately accounts for the true nature of radar reflections. The literature on the physics of radar reflections [10,12] states that the reflections are random, not deterministic. We start with a model which takes into account this randomness (the discrete version of this model is given by problem three above). The processing to obtain the "best" image is then derived [1]. This processing is valid for any transmitted signal as long as the model is valid. Thus, this processing could be used on the radar returns from any system presently implemented if the target falls into the category covered by the model. What we aim to do in this section of the paper is to introduce a procedure for selecting the transmitted signal optimally. The signal is selected to maximize a measure of the achievable performance. The potential use of this procedure in practice and open questions which we are exploring are addressed shortly.

The measure which is used to determine the relative merit of a transmitted signal is the performance which is achievable. The performance is measured in terms of the Fisher information matrix. Other authors have used this measure [6,7,8]. It has been recognized that maximizing a norm on the Fisher information matrix is not always equivalent to minimizing a relevant norm on the Cramer-Rao lower bound. If the lowest eigenvalue of the Fisher information matrix is not made large by the maximization scheme, the largest eigenvalue of the inverse of this matrix may not be small.

Definition: Let S denote the set of admissible signal matrices. Any signal matrix $S \in \mathrm{S}$ defines a matrix Γ_3 from (2.10) and a $P \times N^2$ matrix G as defined in Lemma 4. Let Ω denote the set of possible target images for Σ. Let $\|\cdot\|_P$ be a given norm on P×P matrices. A signal matrix $S \in \mathrm{S}$ is said to be optimal if S achieves the supremum

$$\sup_{S \in \mathrm{S}} \inf_{\Sigma \in \Omega} \| G (K_r^{-1} \otimes K_r^{-1*}) G^\dagger \|_P . \tag{4.1}$$

The three objects which must be given in order to attempt to calculate the optimal S are S, Ω, and $\|\cdot\|_P$. We now attempt a first pass for finding S. First assumptions about Ω are made. Notice that in the definition of the optimal S, the worst possible $\Sigma \in \Omega$ is taken into account. An alternate approach is to assume some a priori density on allowable Σ and then to maximize the expected value of the norm of the Fisher information matrix.

In order to simplify the derivation, we assume that the set of possible target images, Ω, has only one member, Σ. Equivalently, it can be assumed that Σ has been found to be the target image which achieves the infimum in (4.1). For the norm, the trace operator is chosen. Since all of the matrices under consideration are hermitian symmetric and positive semidefinite, the trace is always greater than or equal to zero and is zero only when all of the eigenvalues are zero (and hence the matrix is the zero matrix because it is similar to a zero matrix). The trace is additive so the triangle inequality holds and a scalar times a matrix scales the trace. Thus for the set of positive semidefinite hermitian symmetric matrices, the trace is a valid norm.

There are several possibilities for the set of signal matrices. For both intuitive and analytical reasons it is important to have a constraint on the admissible signal matrices. Intuitively, since the noise intensity does not depend on the signal magnitude, one would expect that increasing the transmitted energy would decrease the effect of the noise. This is the case analytically also. Parameterize the signal matrix as $S = \sqrt{E_t}\hat{S}$, where E_t may be thought of as the transmitted energy. Obviously, Γ_3 may then be written as $\sqrt{E_t}\hat{\Gamma}_3$, where $\hat{\Gamma}_3$ is obtained from Γ_3 by substituting $\hat{S}$ for S. Similarly, from (2.10) and (2.11)

$$K_3 = E_t \hat{\Gamma}_3^\dagger \Sigma \hat{\Gamma}_3 = E_t \hat{K}_3, \tag{4.2}$$

and

$$K_r = E_t \hat{K}_3 + N_0 I_N . \tag{4.3}$$

From $\hat{\Gamma}_3$ form the matrix $\hat{G}$ the same way G was formed from Γ_3. Then the Fisher information matrix (3.5) may be rewritten as

$$\begin{aligned} & E_t^2 \hat{G} [(E_t \hat{K}_3 + N_0 I_N)^{-1} \otimes (E_t \hat{K}_3 + N_0 I_N)^{-1*}] \hat{G}^\dagger \\ &= \hat{G} [(\hat{K}_3 + (N_0/E_t) I_N)^{-1} \otimes (\hat{K}_3 + (N_0/E_t) I_N)^{-1*}] \hat{G}^\dagger . \end{aligned} \tag{4.4}$$

Letting E_t get large decreases the eigenvalues of $\hat{K}_3 + (N_0/E_t) I_N$, increasing the eigenvalues of its inverse. This increases the eigenvalues of the Fisher information matrix and hence increases its trace (the chosen norm). Thus if the signal matrix were unconstrained in size, the optimal signal would be infinite.

There are many ways to constrain the allowed magnitude of S. Suppose that $M : \mathrm{S} \to R^+$ is a measure of how large $S \in \mathrm{S}$ is. Then we may reformulate the problem as maximize the norm of the Fisher information matrix subject to the constraint that $M(S) = E_t$. For example, if the energy is a quadratic function of S, M may be selected to be the trace of $S^\dagger S$. At this point the measure is not too important as long as it constrains the largest value any element of S may take. This is not to say that the computational results will not depend on M, because they will.

Using any standard text on optimization theory (for example, see [13]), a necessary condition for the solution to this problem is that there exist a λ and an $S \in \mathrm{S}$ such that

$$\frac{\partial Tr}{\partial S} [G (K_r^{-1} \otimes K_r^{-1*}) G^\dagger] + \lambda \frac{\partial M(S)}{\partial S} = 0, \tag{4.5}$$

and

$$M(S) = E_t . \tag{4.6}$$

The partial derivative of the trace of the Fisher information matrix with respect to S in the direction δS is

$$Tr\,[K_r^{-1} S^{\dagger} V^{\dagger} V S K_r^{\dagger} (\delta S^{\dagger} V^{\dagger} V S + S^{\dagger} V^{\dagger} V \delta S)], \qquad (4.7)$$

where V is the matrix from (2.9), (2.10), and the accompanying discussion such that $\Gamma_3 = VS$. After selecting a particular M, the previous equations are solved for the signal matrix.

This approach to signal selection has the disadvantage that the equations to be solved are highly nonlinear. It should be noticed that the signal matrix enters the partial derivative trace result both directly and through the matrix K_r^{-1}. Even though these equations may be difficult to solve, in practice they are usually computed offline and thus would not affect the realizability of the proposed system. Any constraints on the signal matrix are included in S and M so any signal computed by the above scheme should produce an implementable signal matrix.

As mentioned previously, the choice of M plays a large part in the complexity of the calculations. A good choice of M could yield a calculation for the signal matrix in closed form.

The results derived so far pertain to the special case where $\Omega = \Sigma$. In an implementation, the actual Σ is unknown. One possible way to take this into account is to transmit multiple pulses. For the first pulse, transmit the signal which would be ideal for some assumed generic target (perhaps a uniform spectrum). Calculate an image using the algorithm in [1]. For pulse k, transmit the pulse which is ideal for the image which is the output of the k-1 imaging stage. The convergence of this scheme must be explored.

5. CONCLUSIONS

This paper has two major results on spectrum estimation problems where the observed data is the sum of white noise and a linear combination of a realization of a portion of a periodic process. The results of each part are derived from the Cramer-Rao lower bound for the variance of spectrum estimates. The first set of results deals with uniqueness of maximum likelihood estimates of the spectrum samples. In particular, if the Fisher information matrix is singular then there is no postitive definite spectrum which yields a unique maximum of the likelihood function. The second set of results proposes a new method for selecting radar signals based on the Fisher information matrix. Transmitted signals are selected to maximize the performance achievable. It is attempted to minimize the variance of the estimates of the spectrum samples. Thus the images produced may be relied upon with more confidence.

We are still in the process of implementing the algorithm proposed in [1] to produce delay Doppler images using maximum likelihood estimation techniques. Some preliminary experiments have been performed on one dimensional versions of the imaging problem and the results appear promising [9]. When going to the full two dimensional problem, the issue of choosing a good transmitted signal arises. It is hoped that the method proposed here will yield better images than methods presently used.

REFERENCES

1. D. L. Snyder, J. A. O'Sullivan, and M. I. Miller, "The Use of Maximum-Likelihood Estimation for Forming Images of Diffuse Radar-Targets from Delay-Doppler Data," *IEEE Transactions on Information Theory,* to appear 1989.
2. M. I. Miller and D. L. Snyder, "The Role of Likelihood and Entropy in Incomplete-Data Problems: Applications to Estimating Point-Process Intensities and Toeplitz Constrained Covariances," *Proc. IEEE, Vol. 75,* No. 7, July 1987.
3. D. Fuhrmann and M. Miller, "On the Existence of Positive Definite Maximum Likelihood Estimates of Structured Covariance Matrices," *IEEE Transactions on Information Theory,* July 1988.
4. M. Miller, D. Fuhrmann, J. O'Sullivan, and D. Snyder, "Maximum-Likelihood Methods for Toeplitz Covariance Estimation and Radar Imaging," to appear in S. Haykin, ed., *Advances in Spectrum Estimation,* Prentice-Hall, 1989.
5. J. Burg, D. Luenberger, and D. Wenger, "Estimation of Structures Covariance Matrices," *Proc. IEEE, Vol. 70,* No. 9, pp. 963-974, September 1982.
6. M. Aoki and R. M. Staley, "On Input Signal Synthesis in Parameter Identification," *Automatica, Vol. 6,* pp. 431-440, 1970.

7. R. K. Mehra, "Optimal Inputs for Linear System Identification," *IEEE Transactions on Automatic Control, Vol. AC-19,* No. 3, pp. 192-200, June 1974.

8. R. K. Mehra, "Optimal Input Signals for Parameter Estimation in Dynamic Systems--Survey and New Results," *IEEE Transactions on Automatic Control, Vol. AC-19,* No. 6, pp. 753-768, December 1974.

9. P. Moulin, D. L. Snyder, and J. A. O'Sullivan, "Performance Analysis for Maximum Likelihood Spectrum Estimation of Periodic Processes from Noisy Data," Electronic Systems and Signals Laboratory internal report, Washington University, St. Louis.

10. H. L. Van Trees, *Detection, Estimation, and Modulation Theory, Part III,* Wiley and Sons, New York, 1971.

11. D. R. Wehner, *High Resolution Radar,* Artech House, Norwood, MA, 1987.

12. J. Shapiro, B. A. Capron, and R. C. Harney, "Imaging and Target Detection with a Heterodyne-Reception Optical Radar," *Applied Optics, Vol. 20,* No. 19, pp. 3292-3313, October 1981.

13. D. Luenberger, *Introduction to Linear and Nonlinear Programming,* Addison Wesley, Reading, MA, 1973.

Robustness in Terms of a J-Divergence Signal-to-Noise Ratio

Douglas J. Warren

Department of Electrical Engineering and Computer Science
University of Illinois at Chicago
Chicago, Illinois 60680

I. Introduction

Due to the frequent difficulty involved in the calculation of probability of error for detectors, alternative measures of detector performance have been routinely sought. Methods that have been used have included bounds on the probability of error (eg. Chernoff bounds), asymptotic measures (asymptotic relative efficiency and efficacy), and suboptimal performance measures such as signal-to-noise ratio and its generalizations. In an earlier work [1], a new class of performance measures was proposed. This class, which includes the generalized signal-to-noise ratio, is based on the Ali-Silvey distance measures for probability distributions [2]. Each of the performance measures was shown to be maximized by the likelihood ratio and to have a desirable robustness property. In this paper, we review and extend that earlier work and we investigate a necessary and sufficient condition for a pair of distributions to be least favorable within a class of distribution pairs for the perfromance measure based on the J-Divergence. In Section II, the new performance measures and their relationship to probability of error are briefly discussed and results on robust design of detectors in terms of the J-Divergence based signal-to-noise ratio are given.

II. J-Divergence Signal-to-Noise Ratio (JDSNR)

For detectors which have Gaussian test statistics it can be shown that, for fixed false alarm probability, the probability of detection is monotonically increasing in the generalized signal-to-noise ratio [3]. In this section, we will define several performance measures for nonlinear detectors with statistics that do not satisfy the central limit theorem. It can be shown that each of these performance measures achieves its maximum for the likelihood ratio detector and that for the likelihood ratio detector the performance measure is equal to one of the Ali-Silvey distances.

The generalized signal-to-noise ratio (SNR) can be written as

$$\mathrm{SNR}_g(\mu_1,\mu_0) = \frac{[\mathrm{cov}_0 gL]^2}{\mathrm{var}_0 g} \tag{1}$$

where L is the likelihood ratio, μ_i is the distribution of the random sequence $\mathbf{X}$ under the hypothesis H_i, and g is a real-valued functional defined on the observation sequence $\mathbf{x}$. If we define the functions $\chi(g) = [g - E_0 g]^2$ and $f(u) = u$ we can rewrite (1) as

$$\mathrm{SNR}_g = f\left\{\frac{[E_0\{\chi^{1/2}(g)\chi^{1/2}(L)\}]^2}{E_0\chi(g)}\right\} \le f\{E_0\chi(L)\} = \mathrm{SNR}_L \tag{2}$$

where we have applied the Cauchy-Schwarz inequality. We can construct a similar inequality for any pair of functions χ and f if $\chi(g)$ and $\chi(L)$ are positive and f is increasing. This leads us to define performance measures for hypothesis testing in terms of distance measures for distributions. Rewriting (2) for an arbitrary distance measure, we have

$$\mathrm{PM}(g;\mu_1,\mu_0) = f\left\{\frac{[E_0\{\chi^{1/2}(g)\chi^{1/2}(L)\}]^2}{E_0\chi(g)}\right\} \le f\{E_0\chi(L)\} = \mathrm{PM}_L(\mu_1,\mu_0) = D(\mu_1,\mu_0) \tag{3}$$

A great many distance measures have been defined for distributions. We will confine the discussion here to the J-Divergence, J; that is, our performance measure is

$$\mathrm{JDSNR}(g;\mu_0,\mu_1) = \frac{\{E_0[(g-1)(L-1)\cdot\log g\cdot\log L]^{1/2}\}^2}{E_0[(g-1)\cdot\log g]} \le E_0[(L-1)\cdot\log L] = J(\mu_1,\mu_0) \tag{4}$$

Clearly, the quantity JDSNR is maximized by the likelihood ratio. It also has some interesting robustness properties. In the following we discuss these properties; Theorem 1 is taken from [1], Theorem 2 is a new result.

Definition. The pair (μ_{0_*},μ_{1_*}) with likelihood ratio $L_* = d\mu_{1_*}/d\mu_{0_*}$ is said to be the least favorable pair of distributions in **P** in terms of the arbitrary performance measure Q if

$$Q(g;\mu_{0_*}{}^n,\mu_{1_*}{}^n) \le Q(L_*{}^n;\mu_{0_*}{}^n,\mu_{1_*}{}^n) \le Q(L_*{}^n;\mu_0{}^n,\mu_1{}^n) \le Q([d\mu_1/d\mu_0]^n;\mu_0{}^n,\mu_1{}^n) \tag{5}$$

for all $(\mu_0,\mu_1) \in \mathbf{P}$. Here the superscripts indicate n-dimensional distributions and likelihood ratios.

Theorem 1. If (μ_{0_*},μ_{1_*}) is the least favorable pair of distributions in **P** in terms of risk then it is also the least favorable pair of distributions in terms of any PM defined by Eq. (3).

Hence we see that robustness in terms of Bayes risk is a sufficient condition for robustness in terms of PM. Unfortunately, it is not, in general, a necessary condition. In [4], it is shown that, if the sets $\mathbf{P}_0$ and $\mathbf{P}_1$ are convex and disjoint, then

$$\operatorname{var}_{0_*} L_* = \min_{(\mu_0,\mu_1)\in \mathbf{P}} \operatorname{var}_0 L$$

is a necessary and sufficient condition for (μ_{0_*},μ_{1_*}) to be the least favorable pair in **P** in terms of SNR. We state a similar result for JDSNR.

Theorem 2. Let $\mathbf{P}_0$ and $\mathbf{P}_1$ be convex, disjoint sets of distributions. A necessary and sufficient condition for (μ_{0_*},μ_{1_*}) to be the least favorable pair of distributions in $\mathbf{P} = \mathbf{P}_0 \times \mathbf{P}_1$ in terms of JDSNR is that $J(\mu_{0_*},\mu_{1_*}) \le J(\mu_0,\mu_1)$ for all $(\mu_0,\mu_1) \in \mathbf{P}$.

Sketch of Proof.

1) Least favorability implies minimum J-Divergence

(μ_{0_*},μ_{1_*}) is least favorable implies that Eq. (5) holds. It can be shown that $J_n = nJ_1$ (subscripts indicate dimension) and, hence, $J(\mu_{0_*},\mu_{1_*}) \le J(\mu_0,\mu_1)$ for all $(\mu_0,\mu_1) \in \mathbf{P}$.

2) Minimum J-Divergence implies least favorability

Let $\nu_i = (1-\alpha)\mu_{i_*} + \alpha\xi_i$, where ξ_i is some distribution in $\mathbf{P}_i$. It can be shown that

$$J(\nu_0,\nu_1) \le (1-\alpha)J(\mu_{0_*},\mu_{1_*}) + \alpha J(\xi_0,\xi_1) \tag{6}$$

for all $\alpha \in [0,1]$. The fact that (μ_{0_*},μ_{1_*}) minimizes the J-Divergence and the convexity property of Eq. (6) together imply that

$$\frac{\partial}{\partial\alpha} J(\nu_0,\nu_1)\,|_{\alpha=0} = E_1 \log L_* - E_0 L_* + E_1(1-L_*{}^{-1}) + E_0(1-L_*) \ge 0 \tag{7}$$

It can also be shown that

$$\mathrm{JDSNR}(L_*{}^n;\nu_0{}^n,\nu_1{}^n) \le (1-\alpha)\,\mathrm{JDSNR}(L_*{}^n;\mu_{0_*}{}^n,\mu_{1_*}{}^n) + \alpha\,\mathrm{JDSNR}(L_*{}^n;\xi_0{}^n,\xi_1{}^n) \tag{8}$$

for all $\alpha \in [0,1]$ and that

$$\frac{\partial}{\partial\alpha} \mathrm{JDSNR}(L_*{}^n;\nu_0,\nu_1)\,|_{\alpha=0} = E_1 \log L_* - E_0 L_* + E_1(1-L_*{}^{-1}) + E_0(1-L_*) \tag{9}$$

Eqs. (7)-(9) imply that $\mathrm{JDSNR}(L_*{}^n;\nu_0,\nu_1)$ is minimized when $\alpha = 0$. Hence, minimum J implies (μ_{0_*},μ_{1_*}) is least favorable for JDSNR which completes the proof.

REFERENCES

[1] D.J. Warren, "Performance criteria for detectors based on distance measures," *Proc. 22nd Conf. Inform. Sci. and Syst.*, pp. 611-615, Princeton University, Princeton, NJ, March 16-18, 1988.

[2] S.M. Ali and S.D. Silvey, "A general class of coefficients of divergence of one distribution from another," *J. Roy. Stat. Soc.*, Ser. B, vol. 28, pp. 131-142, 1966.

[3] D.J. Warren and J.B. Thomas, "Signal-to-noise ratio and central limit theorem considerations in non-Gaussian detection," *Proc. 23rd Allerton Conf. Communications, Control and Computing*, pp. 35-44, Monticello, IL, October, 1985

[4] H.V. Poor, "Robust decision design using a distance criterion," *IEEE Trans. Inform. Theory*, vol. IT-26, pp. 575-587, Sept. 1980.

RESISTANCE AND ROBUSTNESS FOR FINITE MEMORY DETECTORS

KENNETH S. VASTOLA
Rensselaer Polytechnic Institute
Troy, New York 12180

ABSTRACT

Resistance of detection procedures using finite memory nonlinearities is considered for arbitrary observation sequences. Resistance was defined to be a data-dependent version of robustness. Robustness is concerned with distributions rather than data directly. A necessary and sufficient condition is given for a procedure to be resistant at all observation sequences.

It is then seen that this "everywhere resistance" is equivalent to "everywhere robustness", by which we mean that the procedure is robust at every possible distribution on $\mathcal{R}^\infty$ (the collection of all realizations of discrete-time random processes).

These results hold for most practical detection procedures including time-varying, dependent cases and many common array detection techniques. For the special case of memoryless detection, it is shown that these results hold if and only if the gross-error-sensitivity and local-shift-sensitivity parameters are finite.

SUMMARY

Consider the following general structure for a family T of detection statistics $\{T_n\}_{n=1}^{\infty}$ for an arbitrary observation sequence $\underline{x} = (x_1, x_2, x_3, \ldots)$

$$T_n(x_1, \ldots, x_n) = \frac{1}{n} \sum_{i=1}^{n} g_i(x_{i-m}, \ldots, x_i) \tag{1}$$

where each g_i, $i = 1, 2, \ldots, n, \ldots$, is a nonlinearity with memory and m is the (finite) bound on memory length. The class of detectors using a statistic of the form (1) is very large and contains most of those which are commonly used in practice.

For example, generalized memoryless correlator detectors, are included in (1). This form includes optimal detectors for a wide variety of important problems such as asymptotic detection of deterministic or stochastic time-varying signals in independent noise. Further, a number of common array detectors such as the generalized correlator array detector fit the form (1) (see for example [4, 5]).

In [1], Boente, Fraiman and Yohai gave a generalization of Tukey's notion of a statistic T being resistant at an arbitrary observation sequence $\underline{x}$. This is a substantial generalization of Hampel's qualitative robustness (see [1-3]). Intuitively, resistance of a statistic T at $\underline{x}$ means that the value of T at $\underline{y}$ is close to its value at $\underline{x}$ whenever $\underline{y}$ is "close" to $\underline{x}$. The definition of $\underline{y}$ being "close" to $\underline{x}$ used here is an appealing one, especially for signal processing in typical environments such as natural and man-made EMI. It says $\underline{x}$ and $\underline{y}$ are within ϵ if each y_i is within ϵ of x_i except for an ϵ fraction of the indices i (for these indices there is no bound). This is a good model of typical measurement error and impulsive contamination, respectively.

Here we consider the resistance (and robustness) of procedures of the general form (1). We show that a statistic T of the form (1) is everywhere resistant and robust (that is, T, is resistant at every observation sequence $\underline{x}$ and robust at every distribution on $\mathcal{R}^{\infty}$) if and only if the sequence of functions $\{g_i\}_{i=k}^{\infty}$ is (uniformly) bounded and equicontinuous for some k. (Recall that equicontinuity means that each g_i is uniformly continuous using the same ϵ and δ).

As a corollary we consider the basic memoryless statistic (e.g., as is optimal for the constant signal, iid noise case). In this setting, T is everywhere resistant if and only if g is bounded and uniformly continuous. In this case, Hampel's gross-error-sensitivity parameter γ and his local-shift- sensitivity parameter λ can be defined (see [3]). The parameter γ is a numerical measure of the effect on a statistic of a few large deviations, while λ measures the effect of a large number of small errors. For memoryless statistics we show that everywhere resistance and robustness is exactly equivalent to the finiteness of λ and γ. This is a result which does not hold in Hampel's original iid work.

REFERENCES

1. Boente, G., R. Fraiman, and V.J.Yohai, "Qualitative Robustness for Stochastic Processes", *Ann. Stat.*, Vol. 15, No. 3, Sept. 1987, pp. 1293-1312.
2. Martin, R.D. and V.J. Yohai, "Robustness in Time Series and Estimating ARMA Models", Tech. Rept., No. 50, Stat. Dept., Univ. Wash., Seattle, June 1984.
3. Huber, P.J., *Robust Statistics*, Wiley, New York, 1981.
4. Poor, H.V., *An Introduction to Signal Detection and Estimation*, Springer-Verlag, New York, 1988.
5. Kassam, S.A., *Signal Detection in Non-Gaussian Noise*, Springer-Verlag, New York 1988.

DECISION FUSION IN DISTRIBUTED DETECTION WITH UNCERTAINTIES

E. DRAKOPOULOS and C. C. LEE
Department of Electrical Engineering and Computer Science
Northwestern University
Evanston, IL 60208

ABSTRACT

A two–level distributed detection system consisting of a number of local sensors and a fusion center is considered. Each sensor makes a decision regarding an underlying binary–hypothesis testing problem. The local decisions are transmitted to the fusion center where the global decision is derived. Each local sensor has an unknown probability to be jammed or faulty. The resulting binary–hypothesis testing problem is solved using basic concepts of Dempster–Shafer theory which provides a decision making method that discounts the degree of confidence on observations where an uncertainty factor exists. The fusion rule is shown to be the comparison of a weighted sum of local decisions to a threshold. A performance study shows that the proposed decision fusion scheme significantly outperforms the fusion rule which is optimum when there are no jammed or faulty sensors and that it performs almost as well as the optimum fusion rule which requires the exact knowledge of the probability of a sensor being jammed or faulty.

I. INTRODUCTION

Multiple–sensor detection systems have been studied by several researchers recently because they provide higher reliability, higher survivability and larger surveillance area than single sensor systems.

There are basically two approaches in the design of a multiple–sensor detection system. In one approach every sensor transmits its observation data to a fusion center where all the processing takes place. In the other, each local sensor processes its observation data and transmits its decision to the fusion center where the global decision is derived. This latter approach is preferred due to cost, reliability, survivability and communication bandwidth considerations.

In this paper we consider a two–level distributed detection system which consists of a number of local detectors and a fusion center. We consider a binary hypothesis testing problem and assume that the observations of the local sensors (and hence the local decisions) are conditionally independent given either hypothesis.

The problem of designing the fusion center decision rule when the decision rules of the local detectors are known has been studied extensively [1–3]. However no results have been reported for the case where one or more of the local detectors are jammed or faulty. This is an interesting problem since reliability and survivability are considered as major advantages of distributed detection.

Wang and Lee [4] studied the performance of a two level distributed detection system in the context of distributed detection of a slow Rayleigh fluctuating target in additive Gaussian noise, when every sensor has a known probability to be faulty or jammed. The events of failure were considered mutually independent and it was assumed that a jammed or faulty sensor has a probability of 1/2 to offer an incorrect decision.

In our approach we assume that each local detector has an unknown probability ρ_i to be jammed or faulty and a known probability π_{ij} to provide an incorrect decision when jammed or faulty, given that hypothesis H_j is true. The unknown probability ρ_i introduces an uncertainty factor in our decision enviroment. Although this uncertainty can not be treated by Bayesian theory, it can be incorporated in the decision process in the context of Dempster–Shafer theory by discounting the degree of support that each local decision provides for the proposition that H_j, $j = 0, 1$ is the true hypothesis. We assume that the only available information regarding ρ_i is an interval $[\kappa_i\ ,\ \lambda_i]$, where $0 \leq \kappa_i \leq \rho_i \leq \lambda_i < 1$, that contains ρ_i. The degree of uncertainty as to ρ_i is reflected by the length of this interval which is therefore used as "discount rate".

The fusion rule is shown to be the comparison of a weighted sum of local decisions to a threshold, the weights being functions of performance characteristics of the local sensors and the uncertainty interval.

To assess the performance of the proposed decision fusion rule (**R1**), we compare it with that of the fusion rule (**R2**) which is optimum when there are no jammed or faulty sensors and with that of the optimum fusion rule (**R3**) which requires exact knowledge of ρ_i. It is shown that **R1** greatly outperforms **R2** and performs almost as well as **R3**.

In section II we derive the decision fusion rule. In section III we study the performance of the proposed decision scheme. Section IV contains some concluding remarks.

II. A DECISION FUSION RULE BASED ON DEMPSTER–SHAFER THEORY

Consider a binary hypothesis testing problem with hypotheses H_j, $j = 0, 1$. In terms of Dempster–Shafer theory, the set $\Theta = \{H_1, H_0\}$ is called the *frame of discernment*. Let 2^Θ denote the power set of Θ. The set function [6]

$$Bel : 2^\Theta \rightarrow [0, 1]$$

is called a *belief function* if and only if it satisfies the following conditions:

1) $Bel(\phi) = 0$

2) $Bel(\Theta) = 1$

3) for any k

$$Bel\Big(\bigcup_{i=1}^{k} A_i\Big) \geq \sum_{\phi \neq I \subseteq \{1,\ldots,k\}} (-1)^{|I|+1} Bel\Big(\bigcap_{i \in I} A_i\Big) \tag{1}$$

where $|I|$ denotes the number of elements in I

As a special case the belief function Bel for which,

$$Bel(A \cup B) = Bel(A) + Bel(B) \quad \text{when} \quad A, B \subset \Theta \text{ are disjoint,} \tag{2}$$

is called *Bayesian belief function.* In fact, as Bayesian belief function is a special case of belief function, Bayesian theory is a special case of Dempster–Shafer theory [6].

Let Bel_o denote the Bayesian priori over Θ which is assumed to be known. The two–level distributed detection system that we study consists of n local detectors and a fusion center. Let y_i, $i = 1, \ldots, n$, denote the observation vector of the i^{th} sensor. Each detector employs a decision rule $\gamma_i(y_i)$ to make a decision u_i, $i = 1, \ldots, n$, where $u_i = j$ when the detector decides H_j, $j = 0, 1$. We assume that the local decision rules $\gamma_i(\cdot)$ are given and that the observations y_i (and hence the decisions u_i) are conditionally independent given either hypothesis. Each local detector has an unknown probability ρ_i to be jammed or faulty and a known probability π_{ij} to provide an incorrect decision when jammed or faulty, given that hypothesis H_j is true. The local decisions u_i are transmitted to the fusion center where the global decision u is determined.

Define the random variable J_i as follows,

$$J_i = \begin{cases} 1 & \text{if } i^{th} \text{ sensor is jammed or faulty} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

and by assumption,

$$P(J_i = 1 \mid H_1) = P(J_i = 1 \mid H_0) = \rho_i \tag{4}$$

Let $P(u_i = 1 \mid J_i = 0, H_1) = P_{d_i}$ and $P(u_i = 1 \mid J_i = 0, H_0) = P_{f_i}$ denote the probabilities of detection and false alarm, respectively, for local detector i when $J_i = 0$. Then,

$$\begin{aligned} P_{D_i} \triangleq P(u_i = 1 \mid H_1) &= \sum_{J_i} P(u_i = 1 \mid J_i, H_1) P(J_i \mid H_1) \\ &= P(u_1 = 1 \mid J_i = 0, H_1)(1 - \rho_i) + P(u_i = 1 \mid J_i = 1, H_1)\rho_i \\ &= P_{d_i}(1 - \rho_i) + P(u_i = 1 \mid J_i = 1, H_1)\rho_i \end{aligned} \tag{5}$$

The effect of jamming is to switch a decision from $u_i = k$ $(k = 0, 1)$ to $u_i = 1 - k$ with probability π_{ij} under hypothesis H_j.

Therefore,

$$P(u_i = 1 \mid J_i = 1, H_1) = (1 - P_{d_i})\pi_{i1} + P_{d_i}(1 - \pi_{i1}) \tag{6}$$

Substituting (5) into (6) we find that,

$$P_{D_i} = P_{d_i}(1-\rho_i) + \left[(1-P_{d_i})\pi_{i1} + P_{d_i}(1-\pi_{i1})\right]\rho_i \tag{7}$$

Similarly, we obtain

$$P_{F_i} \triangleq P(u_i = 1 \mid H_0) = P_{f_i}(1-\rho_i) + \left[(1-P_{f_i})\pi_{i0} + P_{f_i}(1-\pi_{i0})\right]\rho_i \tag{8}$$

In the fusion center each local decision u_i can be considered as an *independent body of evidence* that provides a degree of support for the proposition that $H_j,\ j = 0,1$ is the true hypothesis. If the probability ρ_i that the i^{th} detector is jammed or faulty is known, the degree of support associated with the local decision u_i can be represented by the belief function Bel_i over 2^Θ which is defined as follows [6]:

$$Bel_i(\{H_j\}) = 1 - \frac{P(u_i \mid H_{1-j})}{\max\limits_{H_j \in \Theta} P(u_i \mid H_j)}, \quad \text{where } j = 0,1 \text{ and } i = 1,\ldots,n. \tag{9}$$

From (7),(8), and (9), we have,

$$Bel_i(\{H_1\}) = \begin{cases} 1 - \frac{P_{F_i}}{\max\{P_{D_i}, P_{F_i}\}} & \text{if } u_i = 1 \\ 1 - \frac{1-P_{F_i}}{\max\{1-P_{D_i}, 1-P_{F_i}\}} & \text{if } u_i = 0 \end{cases} \tag{10}$$

$$Bel_i(\{H_0\}) = \begin{cases} 1 - \frac{P_{D_i}}{\max\{P_{D_i}, P_{F_i}\}} & \text{if } u_i = 1 \\ 1 - \frac{1-P_{D_i}}{\max\{1-P_{D_i}, 1-P_{F_i}\}} & \text{if } u_i = 0 \end{cases} \tag{11}$$

The existence of jammed or faulty detectors introduces an uncertainty factor in our decision environment. Although this uncertainty cannot be treated by Bayesian theory, it can be incorporated in the decision in the context of Dempster–Shafer theory by discounting the degree of support $Bel_i(\{H_j\})$. The discount rate α_i that will be applied to the evidence provided by the local decision u_i must be related to the unknown probability ρ_i that the local detector i is jammed or faulty. Although in practice it is not likely to know the exact value of ρ_i it is possible to determine an interval $[\kappa_i\ ,\ \lambda_i]$, where $0 \le \kappa_i \le \rho_i \le \lambda_i < 1$, that contains the value of ρ_i. Assuming that this is the only available information regarding ρ_i then the most logical estimate of ρ_i is the middle point of the interval, $(\kappa_i + \lambda_i)/2$. Based on this estimate, a fusion rule can be developed, at least in principle. However, the deviation of the estimate from the true ρ_i can be very significant, especially when $(\lambda_i - \kappa_i)$ is large. Apparently the degree of uncertainty as to ρ_i is reflected by $(\lambda_i - \kappa_i)$. Therefore, it is logical to define a discount rate α_i as

$$\alpha_i \overset{\text{def}}{=} \lambda_i - \kappa_i \quad \text{where } 0 \leq \kappa_i \leq \rho_i \leq \lambda_i < 1 \text{ and } i = 1, \dots, n \tag{12}$$

Note that the larger the length of the interval $[\kappa_i\ ,\ \lambda_i]$ the larger the uncertainty in our decision environment and consequently the larger the discount rate α_i. In particular, exact knowledge of ρ_i means a zero length interval and hence a zero discount rate.

Let $\widehat{Bel}_i(\cdot)$ denote the belief function (9) obtained when the point $(\kappa_i + \lambda_i)/2$ is used as an estimate of ρ_i and $Bel_i^{\alpha}(\cdot)$ denote the belief function obtained after discounting the evidence associated with the local decision u_i with a discount rate α_i. Then we have [6],

$$Bel_i^{\alpha}(\{H_j\}) = (1 - \alpha_i)\widehat{Bel}_i(\{H_j\}) \quad \text{where } i = 1, \dots, n, \ \ j = 0, 1 \tag{13}$$

The total degree of support that each $\{H_j\}$ receives can be obtained by combining the belief functions $Bel_i^{\alpha}(\cdot)$, $i = 1, \dots, n$, and the Bayesian priori $Bel_o(\cdot)$ using Dempster's rule of combination [6, chapter 3]:

$$Bel(\{H_j\}) = Bel_o(\{H_j\}) \oplus Bel_t(\{H_j\})$$

where,

$$Bel_t(\{H_j\}) = \bigoplus_{i=1}^{n} Bel_i^{\alpha}(\{H_j\}) = 1 - \prod_{i=1}^{n} \left[1 - Bel_i^{\alpha}(\{H_j\})\right], \tag{14}$$

Then by theorem 9.7 in [6], we obtain

$$Bel(\{H_j\}) = kBel_o(\{H_j\}) \prod_{i=1}^{n} \left[1 - Bel_i^{\alpha}(\{H_{1-j}\})\right], \quad j = 0, 1 \tag{15}$$

where,

$$k = \left[\sum_{j=0,1} Bel_o(\{H_j\}) \prod_{i=1}^{n} \left[1 - Bel_i^{\alpha}(\{H_{1-j}\})\right]\right]^{-1} \tag{16}$$

The global decision will be in favour of the hypothesis H_j that has the larger belief:

$$Bel(\{H_1\}) \underset{u=0}{\overset{u=1}{\gtrless}} Bel(\{H_o\}) \tag{17}$$

Substituting (13), (15) into (17) and taking logarithms on both sides yields,

$$\begin{aligned}\log \frac{Bel_o(\{H_1\})}{Bel_o(\{H_0\})} &+ \sum_{i \in A_1} \log \left[\frac{1-(1-\alpha_i)\widehat{Bel}_i(\{H_0\})}{1-(1-\alpha_i)\widehat{Bel}_i(\{H_1\})} \right] \\ &+ \sum_{i \in A_0} \log \left[\frac{1-(1-\alpha_i)\widehat{Bel}_i(\{H_0\})}{1-(1-\alpha_i)\widehat{Bel}_i(\{H_1\})} \right] \mathop{\gtrless}\limits^{u=1}_{u=0} 0\end{aligned} \tag{18}$$

where $A_j = \{i : u_i = j, 1 \leq i \leq n\}$ and $j = 0, 1$. Equivalently, we have

$$\begin{aligned}\log \frac{Bel_o(\{H_1\})}{Bel_o(\{H_0\})} &+ \sum_{i \in A_1} \log \left[\frac{1-(1-\alpha_i)\left[1-\widehat{P}_{D_i}/\max\{\widehat{P}_{D_i}, \widehat{P}_{F_i}\}\right]}{1-(1-\alpha_i)\left[1-\widehat{P}_{F_i}/\max\{\widehat{P}_{D_i}, \widehat{P}_{F_i}\}\right]} \right] \\ &+ \sum_{i \in A_0} \log \left[\frac{1-(1-\alpha_i)\left[1-(1-\widehat{P}_{D_i})/\max\{1-\widehat{P}_{D_i}, 1-\widehat{P}_{F_i}\}\right]}{1-(1-\alpha_i)\left[1-(1-\widehat{P}_{F_i})/\max\{1-\widehat{P}_{D_i}, 1-\widehat{P}_{F_i}\}\right]} \right] \mathop{\gtrless}\limits^{u=1}_{u=0} 0\end{aligned} \tag{19}$$

where,

$$\begin{aligned}\widehat{P}_{D_i} &= P_{d_i}(1-\hat{\rho}_i) + \left[(1-P_{d_i})\pi_{i1} + P_{d_i}(1-\pi_{i1})\right]\hat{\rho}_i \\ \widehat{P}_{F_i} &= P_{f_i}(1-\hat{\rho}_i) + \left[(1-P_{f_i})\pi_{i0} + P_{f_i}(1-\pi_{i0})\right]\hat{\rho}_i\end{aligned} \tag{20}$$

and $\hat{\rho}_i = (\kappa_i + \lambda_i)/2$.
Therefore the decision fusion rule can be expressed as follows,

$$T = w_o + \sum_{i=1}^{n} w_i \mathop{\gtrless}\limits^{u=1}_{u=0} 0 \tag{21}$$

where

$$w_o = \log \frac{Bel_o(\{H_1\})}{Bel_o(\{H_0\})} \tag{22a}$$

$$w_i = \log \left[\frac{1-(1-\alpha_i)\left[1-\widehat{P}_{D_i}/\max\{\widehat{P}_{D_i}, \widehat{P}_{F_i}\}\right]}{1-(1-\alpha_i)\left[1-\widehat{P}_{F_i}/\max\{\widehat{P}_{D_i}, \widehat{P}_{F_i}\}\right]} \right] \text{ if } u_i = 1 \tag{22b}$$

and

$$w_i = \log\left[\frac{1-(1-\alpha_i)\Big[1-(1-\widehat{P}_{D_i})/\max\{1-\widehat{P}_{D_i},1-\widehat{P}_{F_i}\}\Big]}{1-(1-\alpha_i)\Big[1-(1-\widehat{P}_{F_i})/\max\{1-\widehat{P}_{D_i},1-\widehat{P}_{F_i}\}\Big]}\right] \quad \text{if } u_i = 0 \quad (22c)$$

The decision fusion rule is thus the comparison of a weighted sum of local decisions to a threshold. The weights are functions of the probabilities of $\widehat{P}_{D_i}$ and $\widehat{P}_{F_i}$ defined in (20) and the discount rate α_i. Note that in the absence of jammed or faulty sensors, (i.e., when $\rho_i = \kappa_i = \lambda_i = 0$) then from (20) we find that $\widehat{P}_{D_i} = P_{d_i}$ and $\widehat{P}_{F_i} = P_{f_i}$. Since every local detector satisfies $P_{d_i} > P_{f_i}$ [5] we have from (22) that,

$$w_i = \begin{cases} \log[\frac{P_{d_i}}{P_{f_i}}] & \text{if } u_i = 1 \\ \log[\frac{1-P_{d_i}}{1-P_{f_i}}] & \text{if } u_i = 0 \end{cases} \qquad (23)$$

Note that in this case the fusion rule (21) degenerates to the one obtained by Chair and Varshney [3].

III. PERFORMANCE

To illustrate the performance of the proposed decision fusion rule (**R1**) for a distributed detection system in the presence of uncertainties mentioned above, we compare it with the fusion rule (**R2**) which is optimum when there are no jammed or faulty sensors. Furthermore we compare it with the optimum fusion rule (**R3**) assuming exact knowledge of the nonzero probability of a sensor being jammed or faulty.

We note that (**R2**) is given by (21) and (23) and that (**R3**) is given by (21) with

$$w_i = \begin{cases} \log[\frac{P_{D_i}}{P_{F_i}}] & \text{if } u_i = 1 \\ \log[\frac{1-P_{D_i}}{1-P_{F_i}}] & \text{if } u_i = 0 \end{cases} \qquad (24)$$

where P_{D_i}, P_{F_i} were defined in (7) and (8) respectively.

In our example we consider the binary hypothesis testing problem for n=7 local detectors. We assume that the hypotheses H_j, $j = 0, 1$ are equally probable i.e., $Bel_o(\{H_j\}) = 1/2$. The uncertainty interval $[\kappa_i\ ,\ \lambda_i]$ is the same for all local detectors. The performance characteristics of the local detectors considered are summarized in Table 1.

Table 1
Probability of detection and false alarm
for a seven sensor system

sensor	1	2	3	4	5	6	7
false alarm	.01	.02	.05	.1	.15	.2	.3
prob. of detection	.99	.98	.95	.9	.85	.8	.7

Note that because we assume that the *a priori* probabilities are equally probable, we have chosen the above performance characteristics of the local sensors such that the probabilities of false alarm and miss are equal.
The probability π_{ij} that the local detector i will provide an incorrect decision when jammed or faulty, given that hypothesis H_j is true, is assumed to be equal to 1/2. Then (20) simplifies to the following :

$$\widehat{P}_{D_i} = P_{d_i}(1 - \hat{\rho}_i) + \hat{\rho}_i/2 \tag{25a}$$

and

$$\widehat{P}_{F_i} = P_{f_i}(1 - \hat{\rho}_i) + \hat{\rho}_i/2 \tag{25b}$$

where $\hat{\rho}_i = (\kappa_i + \lambda_i)/2,\ 0 \leq \kappa_i \leq \rho_i \leq \lambda_i < 1$.

Since every detector satisfies $P_{d_i} > P_{f_i}$, we conclude from (25) that $\widehat{P}_{D_i} > \widehat{P}_{F_i}$ for all $i = 1, \ldots, n$. Using this fact the weights w_i defined in (22) can be written as,

$$w_i = -\log\left[1 - (1 - \alpha_i)[\frac{\widehat{P}_{D_i} - \widehat{P}_{F_i}}{\widehat{P}_{F_i}}]\right] \text{ if } u_i = 1 \tag{26a}$$

and

$$w_i = \log\left[1 - (1 - \alpha_i)[\frac{\widehat{P}_{D_i} - \widehat{P}_{F_i}}{1 - \widehat{P}_{F_i}}]\right] \text{ if } u_i = 0 \tag{26b}$$

As performance criterion we use the system's probability of error,

$$P_e = P(u = 1 \mid H_0)P(H_0) + P(u = 0 \mid H_1)(1 - P(H_0))$$

$$\sum_{T \geq 0} \prod_{i=1}^{n} P(u_i \mid H_0)P(H_0) + \sum_{T \geq 0} \prod_{i=1}^{n} P(u_i \mid H_1)(1 - P(H_0)) \tag{27}$$

where T is the test statistic defined in (21).
In Figures 1–4, we plot the system probability of error, for **R1**, **R2**, **R3** versus the probability ρ_i that a sensor is being jammed or faulty, for uncertainty intervals [0.3, 0.4], [0.2, 0.5], [0.05, 0.65], and [0, 0.9], respectively.
From Figure 1 we observe that in the presence of jammed or faulty sensors with uncertainty interval length equal to 0.1 the **R1** significantly outperforms **R2** and

its performance is almost identical to **R3** which requires exact knowledge of ρ_i however.

From Figures 2, 3, and 4, we see that even for large uncertainty intervals **R1** performs significantly better than **R2** and nearly as well as **R3** when the probability of jammed or faulty sensors is significant. For very small values of ρ_i and large uncertainty intervals the performance degradation by using **R1** is very small.

IV. CONCLUSIONS

In this paper we have studied a two–level distributed binary hypothesis detection problem with uncertainties, using basic concepts of Dempster–Shafer theory. We proposed a method of discounting the degree of support provided by each local decision for the proposition that H_j, $j = 0, 1$ is the true hypothesis, when the probability ρ_i of a sensor being jammed or defective is unknown. As discount rate we proposed the length of the uncertainty interval that contains the unknown probability ρ_i. The decision fusion rule was found to be the comparison of a weighted sum of the local decisions to a threshold, the weights being functions of performance characteristics of the local sensors and the uncertainty interval. We showed by performance study that the proposed decision fusion scheme significantly outperforms the fusion rule which is optimum assuming that there are no jammed or faulty sensors and that it performs almost as well as the optimum fusion rule which requires the exact knowledge of the probability of a sensor being jammed or faulty.

REFERENCES

[1] Z. Chair and P.K. Varshney, "Optimum data fusion in multiple sensor detection systems," *IEEE Trans. Aerospace and Electronic systems*, vol. AES-22, pp. 98–101, Jan. 1986.

[2] S.c.A. Thomopoulos, R.Viswanathan and D.P. Bougoulias, "Optimal Decision Fusion in Multiple Sensor Systems," *IEEE Trans. Aerospace and Electronic systems*, vol. AES-23, pp. 644–653, Sep. 1987.

[3] A.R. Reibman and L.W. Nolte, "Design and Performance Comparison of Distributed Detection Networks," *IEEE Trans. Aerospace and Electronic systems*, vol. AES-23, pp. 789–797, Nov. 1987.

[4] T.Y. Wang and C.C. Lee, "Performance of Distributed Detection with Partially Jammed Input," *Proc. of the Twenty-second Annual Conference on Information Sciences and Systems*, Princeton, N.J., March 1988.

[5] H.L. Van Trees, "Detection Estimation and Modulation theory,"John Wiley and sons, NY, 1968

[6] G. Shafer. "A Mathematical Theory of Evidence," Princeton University Press, Princeton, 1976.

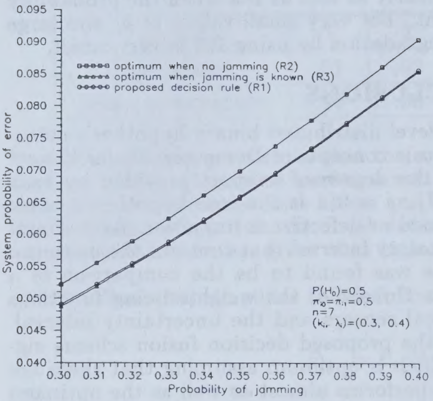


Figure 1. System probability of error vs. probability of a sensor being jammed or defective when the uncertainty interval is (0.3, 0.4) for all sensors.

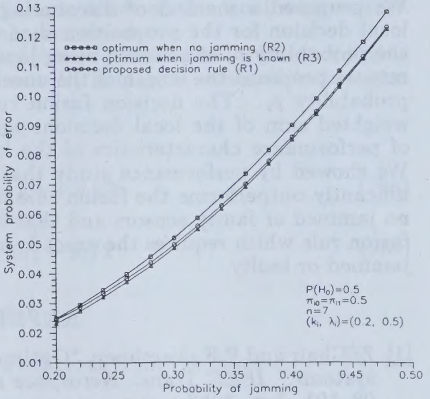


Figure 2. System probability of error vs. probability of a sensor being jammed or defective when the uncertainty interval is (0.2, 0.5) for all sensors.

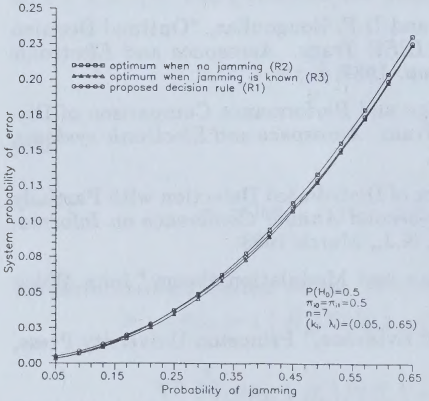


Figure 3. System probability of error vs. probability of a sensor being jammed or defective when the uncertainty interval is (0.05, 0.65) for all sensors.

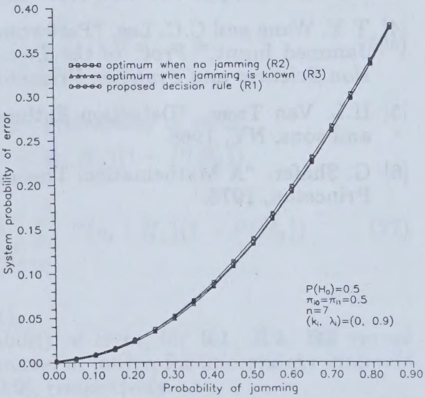


Figure 4. System probability of error vs. probability of a sensor being jammed or defective when the uncertainty interval is (0, 0.9) for all sensors.

THE NUMBER OF MAXIMAL EDGE SEQUENCES ON A CONVEX POLYTOPE

CATHERINE SCHEVON
Department of Computer Science
The Johns Hopkins University
Baltimore, MD 21218

JOSEPH O'ROURKE
Department of Computer Science
Smith College
Northampton, MA 01063

Abstract

We show a tight bound of $\theta(n^3)$ on the number of maximal sequences of edges traversed by shortest paths on the surface of a convex polytope. This result also provides an alternate proof that the total number of edge sequences is $O(n^4)$.

1. INTRODUCTION AND HISTORY

We can group shortest paths on the surface of a convex polytope P by considering two paths to be equivalent when they traverse the same sequence of edges. A *shortest path* between two points x and y on P is a path on the surface of minimum length between x and y. A *shortest path edge sequence* (or "edge sequence") can be defined as an ordered list of edges of P such that any two adjacent edges share a common face, and such that there exists a shortest path traversing the edges in the list. An edge sequence is considered *maximal* iff it cannot be extended to form another valid sequence. The study of these edge sequences is motivated by two problems whose current solutions depend on the number of maximal edge sequences on a convex polytope, as well as on a polynomial-time algorithm to compute them. The first is the problem of computing shortest paths in 3-space amidst convex obstacles, solved by Sharir[5] in a paper in which he also gave an algorithm for computing edge sequences. The second application is the problem of finding the geodesic diameter of a convex polytope. Our algorithm for this problem [3] depends crucially on edge sequences.

Sharir's algorithm for computing edges sequences provided a bound of $O(n^7)$ on the number of edge sequences (and hence the number of maximal edge sequences) of a convex polytope [5]. Mount improved the bound to $O(n^4)$[2] and gave an example to show that this bound is tight [2]. In this paper we show that the number of *maximal* edge sequences is $O(n^3)$ using a graph-theoretic argument, and we give an example similar to Mount's to prove that the bound is tight. Our proof also provides a simpler version of Mount's result.

2. UPPER BOUND

The proof of the upper bound involves converting maximal edge sequences on a polytope P to maximal paths in a planar graph G with certain properties that carry over from the geometric world, and then bounding the number of possible maximal paths. We find this bound by counting in three conceptual layers. First we use an assignment scheme to distribute paths originating at one face of P, say, face i, among the other faces. Each component of the bottommost layer consists of the $O(n)$ paths originating from face i and assigned to face j. This bound is based on Mount's result that there are $O(n)$ non-crossing paths between two nodes [2]. We sum these components over all faces j to obtain a count of $O(n^2)$ on the size of each component of the middle layer, consisting of paths originating from face i. Finally we obtain an $O(n^3)$ bound on the size of the top layer, containing all the maximal paths in the graph G, as the sum of the middle layer components over all faces i of P.

We convert edge sequences on the polytope P to paths in a planar graph by using a representation of the dual of P, as described in [2]. First, we assume that P is triangulated. Let G be the *combinatorial dual* of P, where G is a graph whose nodes correspond to the faces of P and whose arcs connect nodes whose corresponding faces in P share a common edge. Note that G is a polytopal graph, and is therefore simple and planar. Since the faces of P are all triangles, each node of G has exactly three adjacent nodes.

The shortest path edge sequences of P correspond to paths in the graph G, as follows: we map an edge sequence, consisting of an alternating sequence of edges and faces, to a path consisting of alternating nodes and arcs that correspond directly to the faces and edges in the sequence. Since shortest paths on a convex polytope do not pass through a vertex, nor travel along an edge (except in the specific case where the path begins at a vertex and goes to a point on an incident edge) [6], the mapping of a sequence of edges and faces in P to a sequence of arcs and nodes in G is valid.

The paths in G inherit certain properties from the properties of shortest paths in P. First, since shortest paths on a convex polytope cannot self-intersect nor cross any edge more than once [6], the corresponding paths in G are edge-simple and vertex-simple. The second property, that no two paths cross each other more than once, derives from the fact that no two shortest paths can cross more than once [6], and from [2, Lemma 3.2]:

Lemma 1: *If two paths in G corresponding to two shortest paths on P cross k times, then the paths on P cross at least k times.*

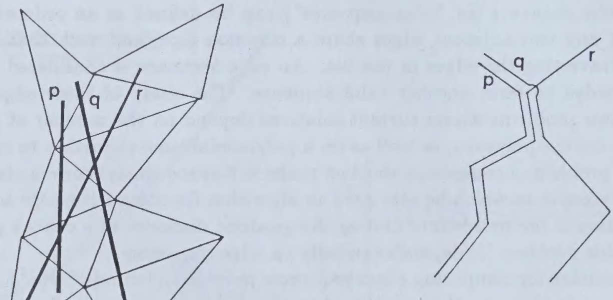


Figure 1: (a) Shortest paths p, q, and r on P. (b) Paths p, q, and r in G.

The notion of crossing edge sequences and paths will be important to the proof, so we will need to make it precise as per Mount's description [2]. Observe that edge sequences (and paths) can share common subsequences without crossing. In Fig. 1, the path p is tangential to paths q and r in G, as they share sequences of faces on the polytope P. Note that the crossing of shortest paths p and q on P does not translate to a crossing of the corresponding paths in the graph G. The paths q and r, on the other hand, are shown to cross in G. In order for two paths p and q to cross at a common subpath, they must enter and exit the subpath on opposite sides (see Fig. 1). Imagine shrinking the shared subpath down to a single vertex v, and examining the four subpaths, p_1, p_2, q_1, and q_2, that result from splitting each path at v.

If their cyclic ordering around v alternates between p and q, the paths cross; otherwise, they intersect tangentially.

We obtain an upper bound on the number of maximal edge sequences on the surface of P by bounding the number of maximal paths in G that obey the following properties:

1. The paths are maximal, i.e. no path is a subpath of any other.

2. The paths are edge- and vertex-simple.

3. No two paths cross each other more than once.

We do this indirectly by bounding the the number of paths that are *half maximal* w.r.t. a given node. It is helpful to consider all paths with one endpoint at a given node to be directed away from that node. Consider the set of paths originating at a given node v, i.e., the sequences corresponding to shortest paths starting at the face corresponding to v on P. A path in the set is half maximal w.r.t. v if it is not a subpath of any other path in the set. From the point of view of v, these are the "longest" paths available: they may not be extended at one end. Note that all maximal paths are half maximal w.r.t. both endpoints, but not all half maximal paths are maximal, since a half maximal path w.r.t. v might be extensible at the v endpoint.

The half maximal paths originating at node v constitute the middle layer in our counting scheme. The following theorem bounds the size of the middle layer by summing the components of the bottom layer.

Theorem 2: *There are $O(n^2)$ half maximal paths starting at a single node.*

Proof: Let $H(v)$ be the set of half maximal paths with an endpoint at node $v \in G$. For a path $a \in H(v)$, let a_k be the initial subpath of a up to the kth node of a. We define $\beta(a)$, the ***branch node*** of a, as the last node of a_k, where k is the largest number such that there is another path $b \in H(v)$ with $b_k = a_k$. Thus $\beta(a)$ is the node at which a first branches from the other paths in $H(v)$, distinguishing itself from them. We also define the node $\delta(a)$ as the ***distinguishing node*** of a, i.e. the one immediately following $\beta(a)$ in a's path. (See Fig. 2.) We ***assign*** each path $a \in H(v)$ uniquely to $\beta(a)$. We will bound the size of $H(v)$ by establishing a bound of $O(n)$ on the number of paths assigned to each node. As all paths in $H(v)$ are assigned to some node, and there are $O(n)$ nodes in G, this will imply the theorem.

Let x be a node other than v, and let X be the set of all paths assigned to x, i.e.

$$X = \{a \in H(v) \mid \beta(a) = x\}$$

Let x_1, x_2, and x_3 be the three nodes adjacent to x. Partition X into three sets depending on which of the three nodes is the first distinguishing node of the paths:

$$X^i = \{a \in H(v) | \beta(a) = x \text{ and } \delta(a) = x_i\},\ i = 1, 2, 3$$

so that a path in X^3, directed from v, exits x via the node x_3.

Clearly the X^i's partition X, so $X = \sum_{i=1}^{3} |X^i|$.

Without loss of generality, we now fix attention on the set X^3. A bound on X^3 will yield a bound on X via the summation given above. Observe that all paths in X^3 have the form $(v, \ldots, x_k, x, x_3)$, where x_k is either x_1 or x_2. We further partition X^3 into the sets X_1^3 and X_2^3 according to whether the path enters x via x_1 or x_2, respectively. Again, since $|X^3| = |X_1^3| + |X_2^3|$, a bound on, say, $|X_1^3|$ will yield the same bound on $|X^3|$ and therefore also on $|X|$.

Let a and b be two distinct paths in the subset X_1^3, so that their sequences follow $(v, \ldots, x_1, x, x_3)$. We designate $x_1 - x$ as the *input* arc into node x and $x - x_2, x - x_3$ as

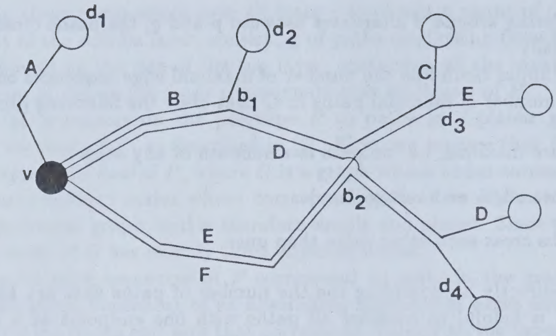


Figure 2: A set of paths and their branch and distinguishing nodes. $\beta(A) = v$, $\beta(B) = b_1$, $\beta(C) = \beta(D) = \beta(E) = \beta(F) = b_2$, $\delta(A) = d_1$, $\delta(B) = d_2$, $\delta(C) = \delta(E) = d_3$, and $\delta(D) = \delta(F) = d_4$.

output arcs, so that the paths we are considering are directed through x via input x_1 and outputs x_2 and x_3.

Because $\beta(a) = x$, there must exist a path $d \in H(v)$ that matches a up to x. Furthermore, since d necessarily diverges from a at x, we have $d \in X_1^2$, so that $d = (v, \ldots, x_1, x, x_2)$. Similarly, let $c \in X_1^2$ be a diverging path for b. The four paths are illustrated in Fig. 3. We have two cases to consider: first, a and b do not cross between v and x, and second, a and b do cross. The former case is that of Fig. 3.

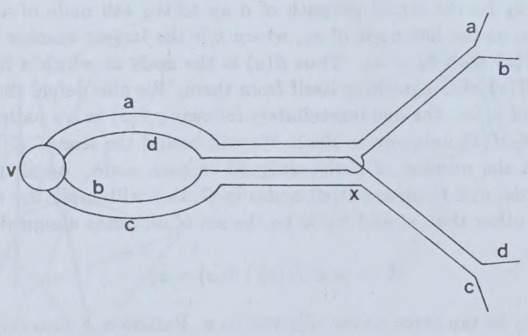


Figure 3: The paths a and b in the set X_1^3 and the paths c and d in X_1^2 crossing at the node x.

Let y be the beginning of the subpath ending at x shared by all four paths. Note that $y \neq v$, since a and b are distinct, but it is possible that $y = x_1$. The four paths enter y along two different edges, with a and d entering on one edge, and b and c entering on the other. Now consider the outputs of node x. Paths a and b pass through x_3 out of x, while the paths c and d pass through x_2. In the first case, where a and b do not cross up to x, we see that the paths b and d must form an alternating sequence in the cyclic ordering about the subpath $y \ldots x$ and therefore cross each other at that subpath. In the second case the same is true of the paths a and c, since the inputs to node y are reversed.

We now show how either case implies that a and b do not cross between v and x. Let $\hat{ad}$ be the initial subpath of a and d up to node y, and similarly define $\hat{bc}$. Since two paths that include $\hat{ad}$ and $\hat{bc}$ as initial subpaths cross each other at the subpath $y \ldots x$, it follows from the property that two paths cross each other at most once that $\hat{ad}$ and $\hat{bc}$ cannot cross between v and y. Therefore, the paths a and b, which also have $\hat{ad}$ and $\hat{bc}$ as initial subpaths, cannot cross each other between v and x, since they cannot cross up to y and they share a common subpath between y and x.

We can now apply [2, Lemma 3.3] to the set X_1^3 to obtain the result that the set X contains $O(n)$ paths:

Lemma 3: *The number of distinct, non-crossing paths between two vertices is* $O(n)$.

Since we have $O(n)$ paths assigned to each of $O(n)$ nodes, we immediately obtain the result that the set $H(v)$ of half maximal paths w.r.t. v is $O(n^2)$ in size. $\square$

We now turn to establishing the size of top layer, consisting of the maximal paths of G, by summing up the components of the middle layer. Since there are $O(n^2)$ half maximal, and therefore maximal, paths at each of the n nodes of G, we have a total bound of $O(n^3)$ on the number of half maximal edge sequences, and thus on the number of maximal edge sequences:

Corollary 4: *There are* $O(n^3)$ *maximal (and half maximal) edge sequences on a convex polytope.*

A bound of $O(n^4)$ edge sequences is obtained by observing that each half maximal sequence w.r.t. node v may have $O(n)$ subsequences with an endpoint at v. Every edge sequence of P is included in this count, since each sequence must be a subsequence (or equal to) a half maximal sequence with respect to each endpoint. This discussion leads immediately to the following two corollaries:

Corollary 5: *There are* $O(n^4)$ *edge sequences on a convex polytope.*

3. LOWER BOUND

We describe a polytope Q that has $\Omega(n^3)$ maximal edge sequences, thereby establishing that Corollary 4 is tight. Q is roughly a cylinder composed of n thin rectangular faces, each with width 1 and height n. For the purpose of simplifying the proof, we will always choose n to be odd. The maximal paths on the cylinder will consist of $\Omega(n)$ "basic" horizontal paths, or paths parallel to the base of the cylinder, and $\Omega(n^2)$ "augmenting" paths for each basic path. The construction of the augmenting paths, in which new edges are added to one face of Q, is the same as that used by Mount in his lower bound example [2]. We first describe the basic paths.

Label the faces of the cylinder F_i, where $i = 1, \ldots, n$. Fig. 4 shows these faces unrolled flat for $n = 7$. Face $F_{\lceil \frac{n}{2} \rceil}$ will be modified later in the construction; all others will remain identical. Let the edge shared by F_i and F_{i+1} be labeled e_i as shown in Fig. 4.

Let $P(i, j)$ be the set of all horizontal paths from F_i to F_j that are midway between the base and the top of the cylinder. Our maximal paths will be drawn from the sets $P(i, i + \lceil \frac{n}{2} \rceil)$ for $i = 1, \ldots, \lfloor \frac{n}{2} \rfloor$. Let M be the union of these sets.

A path $p \in P(i, i + \lceil \frac{n}{2} \rceil)$ realizes the edge sequence $(e_i, e_{i+1}, \ldots, e_{i+\lceil \frac{n}{2} \rceil - 1})$. Each path in M crosses the face $F_{\lceil \frac{n}{2} \rceil}$ and has a length l satisfying $\lfloor \frac{n}{2} \rfloor \leq l < \lfloor \frac{n}{2} \rfloor + 2$.

We now focus on a subset M^* of M, defined as all those paths in M with length $l < \lfloor \frac{n}{2} \rfloor + \frac{1}{4}$. We now argue that every path in M^* (a) is a shortest path, and (b) determines a maximal edge sequence.

Lemma 6: *Every path in* M^* *is shortest.*

Proof: Let p be a path in M^* with endpoints A and B, and length $l < \lfloor \frac{n}{2} \rfloor + \frac{1}{4}$. Because the cylinder has height n, any path from A to B crossing the top or base of the cylinder is considerably longer than l. Thus the only competitor to p is the horizontal path p' around the back of the cylinder. The length l' of p' is $l' = n - l > \lceil \frac{n}{2} \rceil - \frac{1}{4}$. Thus p is shorter than p' by a length $\delta = l' - l > \lceil \frac{n}{2} \rceil - \frac{1}{4} - \lfloor \frac{n}{2} \rfloor - \frac{1}{4} > \frac{1}{2}$. $\square$

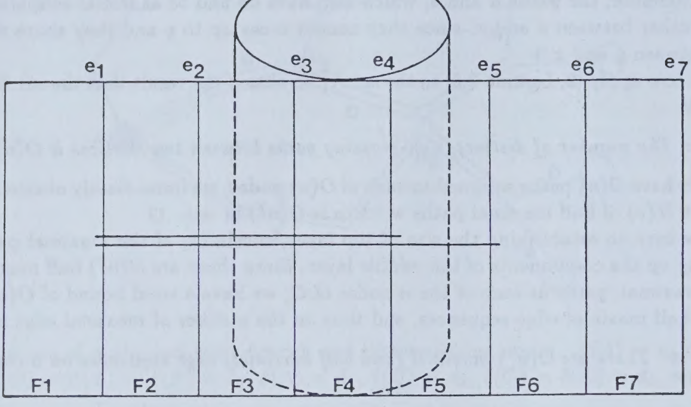


Figure 4: An unrolling of the cylinder for $n = 7$. The vertical scale is reduced for clarity. Two paths are shown: a maximal shortest path in $P(1,5)$, and a path in $P(1,6)$ that is not a shortest path.

Lemma 7: *Every path in M^* realizes a maximal edge sequence.*

Proof: Suppose $p \in M^*$ realizes the edge sequence $E = (e_i, \ldots, e_{i+\lceil\frac{n}{2}\rceil-1})$. Then E is not maximal if there is some shortest path p' that realizes the slightly longer edge sequence $E' = (e_i, \ldots, e_{i+\lceil\frac{n}{2}\rceil})$. But the length l' of such a path p' must satisfy $l' \geq \lceil\frac{n}{2}\rceil$. If p' is horizontal, then the alternative path around the back of the cylinder has length $n - l' < \lfloor\frac{n}{2}\rfloor$, showing that p' is not shortest by $\delta = \lceil\frac{n}{2}\rceil - \lfloor\frac{n}{2}\rfloor = 1$. Paths that are not horizontal similarly lose to the back route. Thus E is a maximal sequence. □

Thus M^* determines a set of $\lfloor\frac{n}{2}\rfloor = \Omega(n)$ maximal edge sequences E^*. We now create $\Omega(n^2)$ "versions" of each sequence by a modification to the construction.

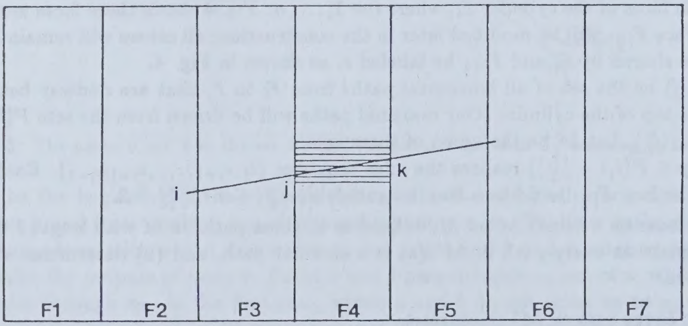


Figure 5: Obtaining n^2 additional paths for each horizontal path. For each i, entry and exit labels j and k may be chosen independently.

Introduce n additional faces in the middle of the face $F_{\lceil\frac{n}{2}\rceil}$ as shown in Fig. 5, labeled

$f_1, \ldots, f_n$. Just as in Mount's lower bound example [2], we can form $\Omega(n^2)$ combinatorially distinct paths from each basic horizontal path through $F_{\lceil \frac{n}{2} \rceil}$ by selecting an entry face f_j and an exit face f_k for each $1 \leq j \leq n$ and $1 \leq k \leq n$. Thus the set of $\Omega(n)$ edge sequences E^* is magnified to a set ES of $\Omega(n^3)$ edge sequences.

We now must argue that all the sequences in ES are (a) shortest path edge sequences, and (b) maximal edge sequences. This is achieved by "squashing" the faces $f_1, \ldots, f_n$ into a height h that is small relative to the cylinder height n. Any path p through f_j and f_k deviates from the horizontal by an angle θ satisfying $\tan\theta < h$, and is lengthened over a truly horizontal path by a factor of at most $\sqrt{1+h^2}$. Because the "slack" δ in the proof of Lemma 6 is an absolute constant $\frac{1}{2}$, choosing h small will ensure that the lemma still holds. Thus the paths are shortest for sufficiently small h. Similarly, Lemma 7, with slack $\delta = 1$, continues to hold for small h.

Therefore all $\Omega(n^3)$ sequences in ES are maximal.

4. ALGORITHM

In this section we describe an $O(n^7 \log n \, 2^{\alpha(n^2)})$ algorithm to compute the shortest path edge sequences of a convex polytope P, improving slightly on Sharir's $O(n^8 \log n)$ algorithm [5]. The method we use is to find the edge sequences of P beginning at each edge e of P by starting at one endpoint of e and travelling along the edge to its other endpoint. We use the algorithm of Mitchell, Mount, and Papadimitriou [1] to compute an initial set of edge sequences at one endpoint of e and then track the changes to this set as the source point moves along the edge. The points at which either a new edge sequence is introduced into the set or an existing sequence becomes invalid, are called *events*. The crux of the algorithm lies in determining the next event point to occur on e and updating current information; its running time depends on the number of these events.

Let t be the parameter of motion of the source point along edge e, so that at the starting endpoint, $t = 0$. Let $e(t)$ be the position of the source point relative to that endpoint. We consider an edge sequence from the source edge e to some edge d to be ***active*** at time t w.r.t. d if there exists a shortest path from $e(t)$ to some point on d that realizes the sequence; otherwise it is ***inactive***. An event on edge d occurs when an active sequence w.r.t. d becomes inactive or when a new or inactive sequence becomes active w.r.t. d. At any time t, there is for each sequence ending on a particular edge d a set of possible events for which it becomes inactive w.r.t. d or active w.r.t. one of the two edges that might follow d in an extended sequence. These potential events are recorded in a priority queue, PQ, of size $O(n^3)$, since there are $O(n^2)$ sequences relevant to a single edge (see Theorem 2).

The edge sequence structure on the polytope P can be viewed as a two dimensional Voronoi diagram of moving points (see Fig. 6). Select a destination edge d and fix it in the plane. We consider the point $d(u)$ on edge d, where u is the motion parameter relative to d in the same sense as t is to edge e. Unfolding the polytope from d back to the source edge e along a particular edge sequence yields an image of e in the plane. The entire set of edge sequences relevant to edges e and d can then be represented as a set of line segments in the plane fixed relative to d, where the source image point $e(t)$ moves along each line segment in a direction determined by the unfolding. The shortest paths from $e(t)$ to the destination edge are constrained to lie in the cells of the Voronoi diagram of the source images, so that the shortest path edge sequence to a particular point on d is determined by which source image is closest to it.

Fig. 6 above shows the position of destination edge d and six source images at time t, labeled e_1 to e_6, moving in the indicated directions.

At time t, there is an event at the point $d(u)$ involving e_3, e_4, and e_6; their Voronoi point touches d at $d(u)$. Just before the event, the Voronoi cells of e_1, e_3, and e_6 intersect the line segment d, making these source images active, while the rest of the images are inactive. When

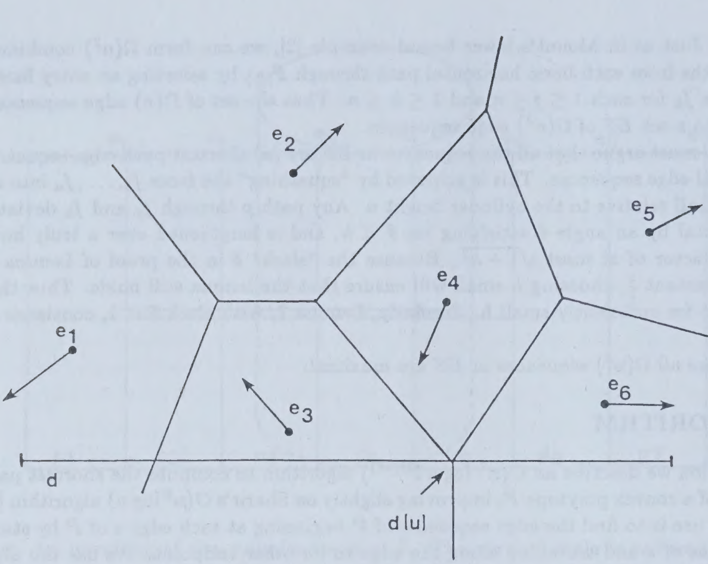


Figure 6: Voronoi diagram of source images.

the Voronoi point has crossed d, e_4's Voronoi cell intersects the line so that e_4 becomes active.

Each edge has associated with it a set of source images and their Voronoi diagram, showing which edge sequences are active and inactive w.r.t. the edge. When an edge sequence first becomes active on an edge d, it is propagated across the next face to the two edges opposite from d, where it becomes a new inactive sequence. In this way we test for single-edge extensions of all known edge sequences. The potential future events associated with all active or inactive source images relevant to each edge are recorded in the queue PQ. The algorithm "jumps" to the next event point by selecting the earliest event from PQ. Let d be the edge on which this event occurs. The changed structure of the Voronoi diagram of the set of source images on edge d at the time of the new event is recorded, and the potential events in PQ are modified as needed. The iteration continues until the edge e has been completely traversed.

The number of event points between the source edge e and a destination edge d can be viewed as the size of the lower envelope of a collection of $O(n^2)$ bivariate functions, each representing one edge sequence. We associate with each sequence E_i beginning on edge e and terminating at edge d a function $f_i(t,u) = dist(e(t); d(u))$, where $dist$ is the Euclidean distance between the two points $e(t)$ and $d(u)$. The shortest path from $e(t)$ to $d(u)$ follows the edge sequence whose function $f_i(t,u)$ has the smallest value. A function $f_i(t,u)$ in the lower envelope at a given time t is active on edge d at that time. The edges of the lower envelope correspond to the boundaries between active source images on edges (as in the edge d in Fig. 6), and the vertices correspond to events, where three source images meet. Therefore, the number of events is bounded by the size of the (planar) diagram of the lower envelope.

The form of the functions f_i satisfies a special case of bivariate lower envelopes studied by Schwartz and Sharir [4] who showed that the size of the lower envelope of N functions in this case is $O(N^2\ 2^{\alpha(N)})$, where $\alpha(n)$ is the inverse Ackerman function and N is the number of functions. Since there are $O(n^2)$ edge sequences between any pair of edges, we have a bound of $O(n^4\ 2^{\alpha(n^2)})$ on the number of events contributed by a particular pair of edges. Taken over all pairs of edges, we obtain a bound of $O(n^6\ 2^{\alpha(n^2)})$ on the number of events on all edges of P.

The overhead of computing the initial configuration at each vertex of P is $O(n^2 \log n)$ per vertex. The main loop of the algorithm runs in $O(n \log n)$ time, with one iteration per event processed. This gives an overall running time of $O(n^7 \log n \, 2^{\alpha(n^2)})$.

5. CONCLUSIONS

We feel that the gap between the number of edge sequences and the time to compute them can be narrowed, but the method employed will have to be rather different from the one outlined above.

The upper bound proof can be generalized to count the number of paths in a planar graph that cross pairwise at most k times. Let $H_k(v)$ be the number of half maximal paths originating at node v and that cross pairwise at most k times. The proof of Theorem 2 essentially reduces $H_k(v)$ to $O(n)$ times the number of paths between two fixed nodes that cross at most $k-1$ times. Since this quantity is bounded by the quantity $H_{k-1}(v)$, we get the following recurrence relation for $H_k(v)$:

$$H_k(v) = n \, H_{k-1}(v), \; k \geq 1$$

and $H_1(v) = O(n^2)$ by Theorem 2. Thus $H_k(v) = O(n^{k+1})$. This provides bounds of $O(n^{k+2})$ on the number of maximal or half maximal paths, and $O(n^{k+3})$ on the total number of paths, for $k \geq 1$.

Acknowledgements

This research was supported by NSF grant DCR-83-51468.

References

[1] J. S. B. Mitchell, D. M. Mount, and C. H. Papadimitriou. The discrete geodesic problem. *SIAM Journal of Computing*, 16:647 – 668, 1987.

[2] D. M. Mount. The number of shortest paths on the surface of a polyhedron. Technical report, University of Maryland, 1987.

[3] J. O'Rourke and C. Schevon. Computing the geodesic diameter of a 3-polytope. manuscript, 1987.

[4] J. Schwartz and M. Sharir. On the two-dimensial davenport schinzel problem. Technical Report 193, NYU Courant Institute, 1987.

[5] M. Sharir. On shortest paths amidst convex polyhedra. *SIAM Journal of Computing*, 16:561 – 572, 1987.

[6] M. Sharir and A. Schorr. On shortest paths in polyhedral spaces. *SIAM Journal of Computing*, 15:193 – 215, 1986.

TWO TECHNIQUES FOR CREATING EXTRA COMPONENTS TO OBTAIN LOWER BOUNDS

PRAKASH RAMANAN
Department of Computer Science
University of California
Santa Barbara, CA 93106

ABSTRACT

In this paper, we consider geometric decision problems whose solution space (set of YES instances) consists of very few connected components. We present two techniques for creating extra components of the solution space, to obtain good lower bounds in the fixed order algebraic decision tree model. The two techniques are fixing some part of the input, and adding some artificial conditions that an YES instance must satisfy. Using these techniques, we present new and simple proofs of a tight $\Omega(n \log n)$ lower bound for computing the width and the diameter of a set S of n points in the plane.

INTRODUCTION

In this paper, we present two techniques to obtain good lower bounds for decision problems whose input instances can be considered as points in the m dimensional Cartesian space E^m. The model of computation we refer to is the *fixed order algebraic decision tree model* [1]. An input instance of a problem P is a sequence $\mathbf{s}$ of m real numbers. $\mathbf{s}$ can be considered as a point in the m dimensional Cartesian space E^m. An algorithm in this model is a binary or ternary decision tree. Each internal node $\mathbf{u}$ of the decision tree contains a test of the form $\mathbf{f_u}(\mathbf{s}):0$, for some polynomial function $\mathbf{f_u}$ of m arguments. For any input $\mathbf{s}$, the computation moves from the root down the decision tree, at each node $\mathbf{u}$ testing and branching according to whether $\mathbf{f_u}(\mathbf{s})$ is < 0, $= 0$ or > 0, until a leaf is reached. Each leaf of the decision tree contains a solution of the problem for each instance that leads to that leaf. The worst-case time-complexity of the algorithm is the height of the decision tree. The phrase *fixed order* refers to the fact that the degrees of the polynomials $\mathbf{f_u}$ are bounded by a constant independent of m.

For a decision problem P whose instances are points in E^m, let the "solution space" W be the set of all YES instances of P. Then P is equivalent to the following *membership problem* for W: Given $\mathbf{s} \in E^m$, determine if $\mathbf{s} \in W$. One well-known technique for obtaining lower bounds for decision problems is based on the following theorem of Ben-Or [1].

Theorem 1. Let $W \subseteq E^m$, and let $\#(W)$ be the number of disjoint connected components of W. Then any fixed order algebraic decision tree that solves the membership problem for W has $\Omega(\log \#(W) - m)$ height.

This theorem can be used to obtain a reasonable lower bound for P if $\#(W)$ or $\#(\overline{W})$ is large, where $\overline{W} = E^m - W$ is the solution space of $\overline{P}$, the complement of problem P. Unfortunately, for some problems, both $\#(W)$ and $\#(\overline{W})$ are small, and the lower bounds obtained using the above theorem are unsatisfactory. For example, we can associate the following decision problem with the problem of computing the width.

This research was supported by NSF under grant CCR-88-03241

WIDTH. Let w be a fixed positive real number. Given a set $S = \{p_1, p_2, ..., p_n\}$ of n points in the plane, determine if $width(S) \leq w$.

Given a YES instance S of WIDTH (a point in E^{2n}), it can be continuously transformed to any other YES instance S' such that all the intermediate instances are YES instances. Hence the solution space W of WIDTH consists of only one connected component. An analogous result holds for $\overline{W}$.

Ramanan [6] proposed the technique of creating artificial components of the solution space, in order to obtain good lower bounds. The technique is to modify the problem P (or $\overline{P}$) slightly, by adding some Artificial Conditions that an YES instance must satisfy, thereby obtaining the problem P_{AC}. Let W_{AC} be the solution space of P_{AC}. The artificial conditions should be such that $\#(W_{AC})$ is large, and P_{AC} is $o(\log\#(W_{AC}))$-time transformable to P. Then, by Theorem 1, we can conclude that P has an $\Omega(\log\#(W_{AC}) - n)$ lower bound. Using this technique, he obtained tight lower bounds for convex polygon inclusion and max gap problems (Lee and Yu [5] had independently obtained a similar result for the max gap problem). Kapoor and Ramanan [4] used this technique, and the technique of fixing part of the input to obtain tight lower bounds for maximal and convex layers problems. In the next section, we illustrate the application of these two techniques to obtain simple, new proofs of an $\Omega(n\log n)$ lower bound for computing the width and diameter.

LOWER BOUNDS FOR COMPUTING THE WIDTH AND DIAMETER

Let S be a set of n points in the plane. The *width* and *diameter* of S, denoted by $width(S)$ and $diam(S)$ respectively, are defined to be the minimum and maximum distance between parallel lines of support of S. Note that $width(S) = width(CH(S))$ and $diam(S) = diam(CH(S))$, where $CH(S)$ is the convex hull of S. Width and diameter have applications in motion planning and in approximating polygonal curves. Brown [2] and Shamos [7] presented $O(n\log n)$ algorithms for computing the diameter of S. Based on these algorithms, Houle and Toussaint [3] presented two $O(n\log n)$ algorithms for computing the width. All these algorithms first find $CH(S)$, and then compute the maximum or minimum distance between parallel lines of support of $CH(S)$. Brown [2] and Lee and Yu [5] proved $\Omega(n\log n)$ lower bounds for computing the diameter and width respectively.

In this section, we present new and simple proofs of an $\Omega(n\log n)$ lower bound for computing the width and diameter, using the techniques mentioned above. Similar to WIDTH, we can associate the following decision problem with the problem of computing the diameter.

DIAM. Let w be a fixed positive real number. Given a set $S = \{p_1, p_2, ..., p_n\}$ of n points in the plane, determine if $diam(S) \leq w$.

Since the solution space W of WIDTH as well as DIAM consists of only one connected component, we create the required number of components using the following two techniques from Ramanan [6] and Kapoor and Ramanan [4]: fixing some part of the input, and adding some Artificial Conditions that an YES instance must satisfy. We also need the following result that has been used to obtain the lower bounds in Kapoor and Ramanan [4], and by other authors.

Lemma 1. Let P be any decision problem whose input is a set $S = \{p_1, p_2, ..., p_n\}$ of n points in E^d. Let R_i, $1 \leq i \leq t$, be disjoint regions in E^d such that in any YES instance of P, each p_i must be in one of these regions. Two YES instances S and S' of P are said to be distinct if $p_i \in S$ is in region R_j, and $p'_i \in S'$ is in region $R_{j'}$, $j \neq j'$. Let $f(n)$ be the number of distinct YES instances of P. Then any fixed order algebraic decision tree that solves P has $\Omega(\log(f(n)) - nd)$ height.

Proof. Let S and S' be two distinct YES instances; let $p_i \in S$ be in region R_j, and $p'_i \in S'$ be in region $R_{j'}$, where $j \neq j'$. Consider any continuous transformation in E^{nd} of (the point representing) the instance S to the instance S'. There must exist an intermediate instance in which the ith input point has moved out of R_j, but is not in any of the regions R_k. By our assumption, this is a NO instance of P. This shows that distinct YES instances correspond to different components of the solution space of P. Hence the solution space W consists of at least $f(n)$ components, and the result follows by Ben-Or's theorem. ■

We use the fact that for a regular polygon, there exists a circumcircle that passes through all its vertices. First, let us consider the problem WIDTHAC (WIDTH with Artificial Conditions).

WIDTHAC. Let RP be a fixed regular convex polygon with the sequence of vertices $(v_1, v_2, ..., v_n)$. Let $w = width(RP)$, and let C be the circumcircle of RP (see Figure 1). Given a set $S = \{p_1, p_2, ..., p_n\}$ of points in the plane, determine if $width(RP \bigcup S) > w$, and if S satisfies the following *Artificial Condition*: All the points in S lie strictly inside C.

Clearly, WIDTHAC is linear-time transformable to the problem of computing the width. Also note that RP, w and C are fixed. An instance of WIDTHAC consists only of S. We prove the following result.

Lemma 2. Any fixed order algebraic decision tree that solves WIDTHAC has $\Omega(n \log n)$ height.

In order to prove this lemma, we need the following definitions and results. A pair of edges of a convex polygon CP is called an *antipodal edge-edge pair (E-E pair)* if there exists a pair of parallel lines of support of CP that contain these edges. *Antipodal vertex-vertex pair (V-V pair)* and *antipodal vertex-edge pair (V-E pair)* are defined similarly. Since a supporting line of CP must touch CP either at a single vertex, or along one of its edges, these are the only three possibilities for antipodal pairs. The following result is well-known.

Lemma 3. The width of a convex polygon CP is the minimum distance between parallel lines of support passing through an antipodal V-E pair.

If L_1 and L_2 are parallel lines, let $d(L_1, L_2)$ denote the distance between L_1 and L_2. A *strip* is part of the plane that is delimited by a pair of parallel lines. Clearly, $width(S)$ is the width of the thinnest strip of the plane that contains S. We have the following converse of Lemma 3.

Lemma 4. Let CP be a convex polygon that is contained in the strip delimited by parallel lines L_1 and L_2. If L_1 and L_2 together do not contain an antipodal V-E pair of CP, then $d(L_1, L_2) > width(CP)$.

Proof. Let $(v_1, v_2, ..., v_n)$ be the sequence of vertices of CP. Let (v_i, v_j) be a pair of vertices such that $v_i \in L_1$ and $v_j \in L_2$; if there is no such pair of vertices, then the result follows. Let α be the smaller of the two angles between L_1 and the line containing v_i and v_j. Suppose L_1 and L_2 do not contain an antipodal V-E pair of CP. Then $(L_1 \bigcup L_2) \bigcap CP = \{v_i, v_j\}$. So, if we rotate CP about v_i by a small amount in the direction of decreasing α, the resulting polygon (which still has width $width(CP)$), still lies between L_1 and L_2, but $L_2 \bigcap CP = \phi$. Hence $d(L_1, L_2) > width(CP)$. ■

Let RP be a regular convex polygon with the sequence of vertices $(v_1, v_2, ..., v_n)$, where n is odd. Let C be the circumcircle of RP. For $1 \leq j \leq n$, the edge e_j opposite a vertex v_j is defined to be the edge of RP that intersects the line passing through v_j and the center of C (see Figure 1); clearly, (v_j, e_j) is an antipodal V-E pair, every antipodal V-E pair in RP is of this form, and $width(RP)$ is the distance of v_j from e_j. For $1 \leq j \leq n$, let R_j be the small open region adjacent to e_j, strictly inside C, and strictly outside RP. Lemma 4 leads to the following result.

Lemma 5. Let RP be a regular convex polygon with the sequence of vertices $(v_1, v_2, ..., v_n)$, where n is odd. Let C be the circumcircle of RP. Let $S = \{p_1, p_2, ..., p_n\}$ be a set of points strictly inside C. Then $width(S \bigcup RP) > width(RP)$ if and only if each of the regions R_j contains one p_i.

Proof. Suppose that for some j, the region R_j does not contain any point of S. Then (v_j, e_j) is an antipodal V-E pair that determines a pair of parallel lines of support of $S \bigcup RP$. Hence $width(S \bigcup RP) = width(RP)$. If each R_j contains a point p_i, then by Lemma 4, every strip of width $width(RP)$ that contains RP fails to contain some point of S. Hence $width(S \bigcup RP) > width(RP)$. ■

Now we are ready to prove Lemma 2.

Proof of Lemma 2. By Lemma 5, each YES instance of WIDTHAC corresponds to an assignment of p_is to the R_js such that the p_is are assigned to different R_js. Hence there are $n!$ distinct YES instances. The result follows from Lemma 1. ■

Since WIDTHAC is linear-time transformable to the problem of computing the width, we have the following result.

Theorem 2. Any fixed order algebraic decision tree that given a planar point set S, $|S| = n$, computes the width of S has $\Omega(n \log n)$ height.

Brown [2] showed that the SET DISJOINTNESS problem (given two sets A and B of n real numbers each, determine if $A \bigcap B = \phi$) is linear-time transformable to the problem of computing the diameter. Since the SET DISJOINTNESS problem has an $\Omega(n \log n)$ lower bound (see [1]), the same lower bound holds for computing the diameter. It doesn't seem likely that a similar approach can be used to obtain a lower bound for computing the width. On the other hand, the problems of computing the width and diameter are very similar. So, we briefly mention the following lower bound proof for computing the diameter that is similar to the one we presented for computing the width. Consider the following problem.

DIAMAC. Let RP be a fixed regular convex polygon with the sequence of vertices $(v_1, v_2, ..., v_n)$. Let $d = diam(RP)$, and let DRP be the largest region in the plane such that $RP \subseteq DRP$, and $diam(DRP) = d$. Let C' be the largest circle that can be inscribed in DRP. Given a set $S = \{p_1, p_2, ..., p_n\}$ of points in the plane, determine if $diam(RP \bigcup S) \leq d$, and if S satisfies the following *Artificial Condition*: All the points in S lie strictly outside C'.

For a regular polygon RP, both DRP and C' are uniquely defined. For even n, $DRP = C'$ is the circumcircle of RP, and so DIAMAC has no YES instances. For odd n, the boundary of DRP consists of n arcs, where each arc is part of a circle centered at one of the vertices of RP, with radius $diam(RP)$. The set of all points that lie strictly outside C' but inside DRP consists of n disjoint regions (one adjacent to each v_j). An instance S of DIAMAC is an YES instance if and only if each p_i lies in one of these regions. Hence, an $\Omega(n \log n)$ lower bound can be obtained as in Lemma 2. Since DIAMAC is linear-time transformable to the problem of computing the diameter, we have the following result.

Theorem 3. Any fixed order algebraic decision tree that given a planar point set S, $|S| = n$, computes the diameter of S has $\Omega(n \log n)$ height.

REFERENCES

1. M. Ben-Or, Lower Bounds for Algebraic Computation Trees, *Proc.* 15th *Ann. ACM Symp. on Theory of Comput.*, May 1983, pp. 80-86.
2. K. Q. Brown, Geometric Transforms for Fast Geometric Algorithms, Ph. D. Thesis, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Pennsylvania, 1979.
3. M. Houle and G. Toussaint, Computing the Width of a Set, *Proc.* 1st *Ann. ACM Symp. on Computational Geometry,* June 1985, pp. 1-7.
4. S. Kapoor and P. Ramanan, Lower Bounds for Maximal and Convex Layers Problems, *Algorithmica,* to appear.
5. D. T. Lee and Y. F. Yu, Geometric Complexity of Some Location Problems, *Algorithmica* **1** (1986), pp. 193-212.
6. P. Ramanan, Obtaining Lower Bounds Using Artificial Components, *Inform. Process. Lett.* **24** (1987), pp. 243-246.
7. M. I. Shamos, Computational Geometry, Ph. D. Thesis, Yale University, New Heaven, Connecticut, 1978.

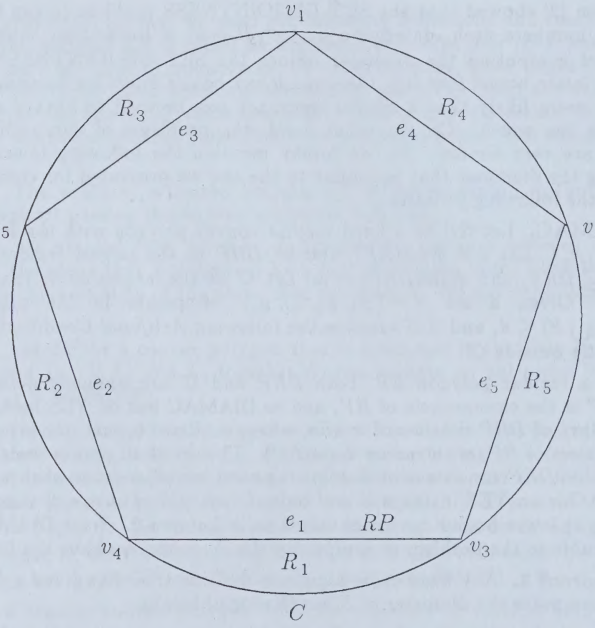


Figure 1

A SEARCH PROBLEM IN PLANE GEOMETRY

KAZUO SUGIHARA Department of Information and Computer Sciences
University of Hawaii at Manoa, Honolulu, HI 96822
ICHIRO SUZUKI Department of Electrical Engineering and Computer Science
University of Wisconsin–Milwaukee, Milwaukee, WI 53201
MASAFUMI YAMASHITA Department of Electrical Engineering, Faculty of Engineering
Hiroshima University, Higashi–Hiroshima 724, Japan

Abstract We consider the problem of searching for a mobile robber in a closed simple two-dimensional region by a number of searchlights. A searchlight is a stationary point which emits a single ray. The ray cannot penetrate the boundary of the region, but its direction can be changed continuously. A point is detected at a given time iff it is on the ray of a searchlight. A robber is a point which can move continuously with unbounded speed. First, we show that the problem of obtaining a search schedule for an instance having at least one searchlight on the boundary of a given region can be reduced to that for instances having no searchlight on the boundary. The reduction is achieved by recursive applications of a simple search strategy called One-Way Sweep Strategy. Then we present various conditions for the existence of a search schedule for instances having two or three searchlights none of which are on the boundary of a given region. Additional sufficient conditions for the existence of a search schedule for instances having an arbitrary number of searchlights are also presented.

1 Introduction

We consider the problem of searching for a mobile robber in a closed simple two-dimensional region by a number of searchlights. A searchlight is a stationary point which emits a single ray. The ray cannot penetrate the boundary of the region, but its direction can be changed continuously. A point is detected at a given time iff it is on the ray of a searchlight. A robber is a point which can move continuously with unbounded speed. We refer to this problem as the *Searchlight Scheduling problem.* The objective is to decide whether there exists a search schedule for detecting a robber regardless of its movement, for a given instance. A possible application of the Searchlight Scheduling problem is security enforcement in industrial plants where searchlights or TV cameras are used to find an intruder.

In the Searchlight Scheduling problem, the locations of searchlights are given as part of a problem instance. Obviously, there exists a search schedule for an instance only if every point in the given region is visible from at least one searchlight. The problem of obtaining a set of locations of searchlights having this property is known as the Art Gallery problem [1] [2] [3] [4].

The first result of this paper is a simple search strategy called One-Way Sweep Strategy. We show that the strategy can be used recursively to reduce the problem of obtaining a search schedule for an instance having at least one searchlight on the boundary of a given region, to that for instances having no searchlight on the boundary. We also present various sufficient conditions for a successful generation of a complete search schedule based on the One-Way Sweep Strategy.

Next, we consider the case in which no searchlight is located on the boundary of a given region. In particular, we present a simple necessary and sufficient condition for the existence of a search schedule for instances having exactly two searchlights. Then we consider the case with three searchlights, and present a number of sufficient conditions for the existence of a search schedule. Some of the conditions we have obtained for this case are quite complex.

It is not a goal of this paper to investigate the computational complexity of the problem assuming, for example, that a region is given as an n-sided polygon. We also note that to our knowledge, the Searchlight Scheduling problem has not been addressed in the literature.

2 Problem Formulation

We use the following notations. The boundary of a two-dimensional region R is denoted by $b(R)$. If R is a simple region, then for points $x, y \in b(R)$, $[x, y]_{b(R)}$ (or $(a, b)_{b(R)}$) denotes the closed (or open) continuous segment of $b(R)$ from x to y taken counterclockwise.

Formally, an instance of the Searchlight Scheduling Problem is a pair $P = (D, L)$, where D is a closed simple two-dimensional region and L is a set of points $l \in D$ called *searchlights*. A point x is said to be *visible* from a searchlight l iff $\overline{lx} \subseteq D$. Note that x is visible from l if $\overline{lx} \subseteq D$, regardless of whether (1)

$l' \in \overline{lx}$ for some searchlight $l' \in L - \{l\}$ or (2) $\overline{lx} \cap b(D) \neq \phi$. We denote by V_l the set of points visible from l.

Definition 1 A *schedule* of a searchlight $l \in L$ is a continuous function f_l from $[0, T]$ to $\mathcal{R}$, where $[0, T]$ is an interval of real time and $\mathcal{R}$ is the set of real numbers. The *ray* of l at time t is the intersection of V_l and the semi-infinite ray with direction $f_l(t)$ emanating from l. (The value of $f_l(t)$ is taken in radian. Directions are measured counterclockwise from the positive x-axis.) We say that l is *aimed at* a point x at time t iff x is on the ray of l at t.

Definition 2 A *trajectory function* r is a continuous function from real time to D. We say that r is *detected* at time t if there exists a searchlight $l \in L$ such that l is aimed at $r(t)$ at t.

Definition 3 $F = \{f_l \mid f_l : [0, T] \to \mathcal{R}$ is a schedule of $l \in L\}$ is a *search schedule* for P if for any trajectory function r, there exists some $0 \leq t \leq T$ such that r is detected at time t.

A trajectory function r represents the trajectory of a robber. Definitions 1, 2 and 3 reflect our intention that (1) at any moment only those points which are on the ray of a searchlight are visible, and (2) a robber can move with unbounded speed.

The following definition provides another interpretation of the process of searching.

Definition 4 A region $R \subseteq D$ is said to be *clear* at time t if, for any trajectory function r such that $r(t) \in R$, there exists some $0 \leq t' \leq t$ such that r is detected at time t'. R is said to be *contaminated* if it is not clear.

Intuitively, R is clear at time t if R does not contain a robber which has not been detected. By definition, D is contaminated initially (at time zero) and becomes clear when the execution of a search schedule is completed.

Example 1 Figure 1(a) shows an instance of the problem with three searchlights l_1, l_2 and l_3. The following is a search schedule. (We use phrases such as "aim l_i at a point x" and "turn l_i clockwise," instead of specifying a function f_{l_i} for l_i explicitly.)

1. Aim l_2 at a.
2. Aim l_3 at a and turn it counterclockwise until it is aimed at b (Figure 1(b)).
3. Aim l_1 at b and turn it counterclockwise until it is aimed at c (Figure 1(c)).
4. Turn l_3 counterclockwise until it is aimed at d (Figure 1(d)).
5. Aim l_1 at g.
6. Turn l_2 clockwise until it is aimed at h (Figure 1(e)).
7. Turn l_1 counterclockwise until it is aimed at h (Figure 1(f)).
8. Turn l_1 clockwise until it is aimed at g (Figure 1(g)).
9. Turn l_3 counterclockwise until it is aimed at e (Figure 1(h)).
10. Aim l_2 at e and turn it counterclockwise until it is aimed at f.
11. Turn l_3 counterclockwise until it is aimed at g (Figure 1(i)).

□

Example 2 Theorem 5 states that there exists no search schedule for the instance shown in Figure 2.
□

Throughout this paper we assume that any given instance $P = (D, L)$ satisfies the following conditions (P1) and (P2), since obviously, otherwise there cannot exist any search schedule.

(P1) $D = \bigcup_{l \in L} V_l$. (Every point in D is visible from at least one searchlight.)

(P2) For each $l \in L$, either $l \in b(D)$ or $l \in V_{l'}$ for some $l' \in L - \{l\}$. (Every searchlight is either on the boundary of a given region or visible from another searchlight.)

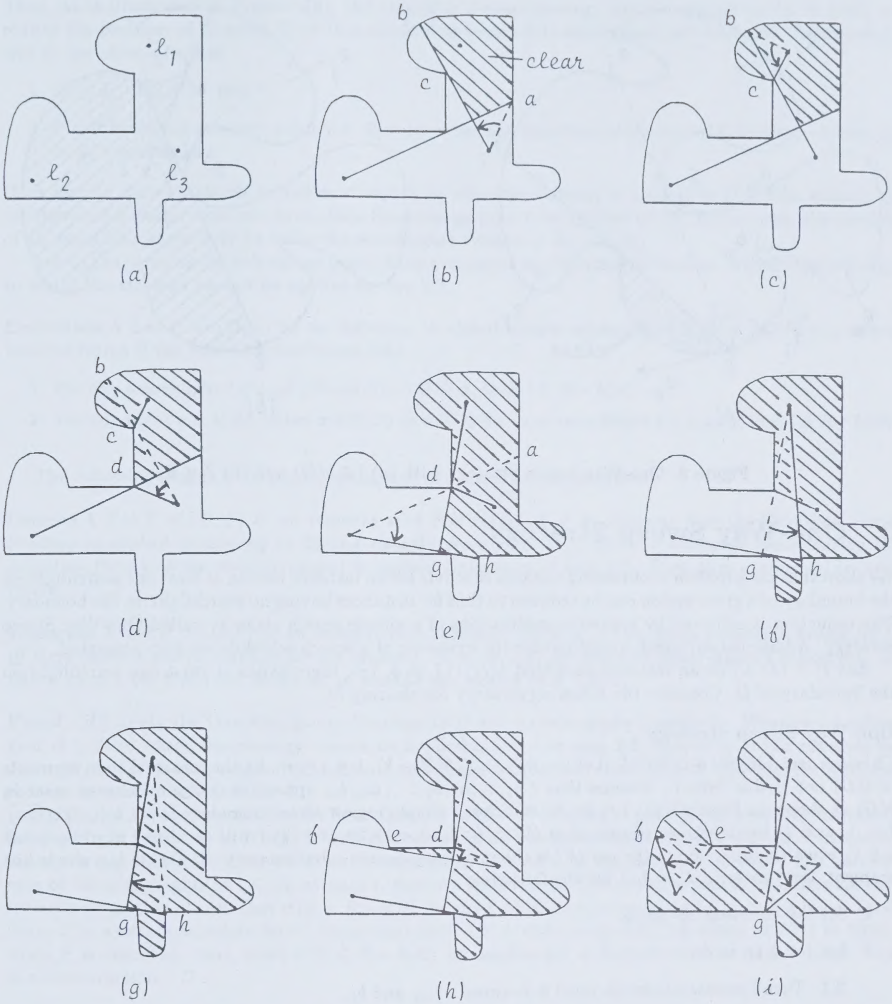

ℓ1
ℓ2
ℓ3
(a)
b
c
clear
a
(b)
b
c
(c)
b
c
d
(d)
a
d
g
h
(e)
g
h
(f)
g
h
(g)
f
e
d
(h)
f
e
g
(i)

Figure 1: A search schedule for an instance of the Searchlight Scheduling problem.

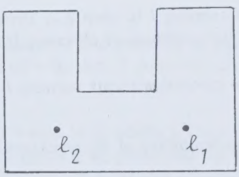


Figure 2: An instance having no search schedule.

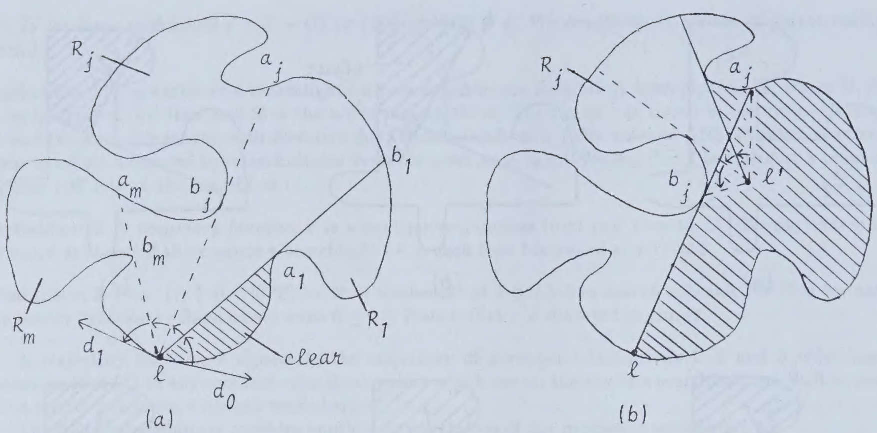


Figure 3: One-Way Sweep Strategy with (a) $l \in b(D)$ and (b) $l' \notin R_j$.

3 One-Way Sweep Strategy

We show that the problem of obtaining a search schedule for an instance having at least one searchlight on the boundary of a given region can be reduced to that for instances having no searchlight on the boundary. The reduction is achieved by recursive applications of a simple search strategy called One-Way Sweep Strategy. Additional sufficient conditions for the existence of a search schedule are also presented.

Let $P = (D, L)$ be an instance such that $b(D) \cap L \neq \phi$, i.e., there exists at least one searchlight on the boundary of D. Consider the following strategy for clearing D.

One-Way Sweep Strategy

Choose a searchlight $l \in b(D) \cap L$. Let $(a_j, b_j)_{b(D)} \subseteq b(D) - V_l$, $1 \leq j \leq m$, be the maximal open segments of $b(D)$ not visible from l. Assume that $l, a_1, b_1, a_2, b_2, \ldots, a_m, b_m$ appear in counterclockwise order in $b(D)$ as shown in Figure 3(a). Let R_j be the closed simple region whose boundary is $[a_j, b_j]_{b(D)} \cup \overline{b_j a_j}$. Let d_0 and d_1 be directions of such that (1) $d_0 \leq d_1 < d_0 + 2\pi$ and (2) l will be aimed at every point $x \in V_l$ when the direction of the ray of l is changed from d_0 to d_1 continuously. (If $D \cap V_l$ is a single line segment, then we have $d_0 = d_1$.) Do the following.

```
1. Set the direction of $l$ to $d_0$.
2. for $j = 1$ to $m$ do
    2.1. Turn $l$ counterclockwise until it is aimed at $a_j$ and $b_j$.
    2.2. Clear $R_j$ without turning $l$. (It may or may not be possible to do this.)
3. Turn $l$ counterclockwise until its direction becomes $d_1$.
```

□

If every R_j can be cleared without turning l in step 2.2, then D becomes clear when step 3 of the strategy is completed. In this sense, the problem of clearing D has been reduced to that of clearing subregions R_j, $1 \leq j \leq m$.

Let us consider how each R_j can be cleared without turning l in step 2.2. Suppose that there exists a searchlight l' such that

1. $l' \notin R_j - b(R_j)$ (l' is either on the boundary of R_j or external to R_j) and
2. $(R_j - b(R_j)) \cap V_{l'} \neq \phi$ (at least one point in the interior of R_j is visible from l').

Then, as is illustrated in Figure 3(b), the One-Way Sweep Strategy can be applied to R_j in order to reduce the problem of clearing R_j to that of clearing some of its subregions, provided that directions d_0 and d_1 are chosen so that

1. $d_0 \leq d_1 < d_0 + 2\pi$ and
2. l' will be aimed at every point $x \in R_j \cap V_{l'}$ when the direction of the ray of l' is changed from d_0 to d_1 continuously.

(It is easy to show that such d_0 and d_1 always exist when the strategy is applied to D and its subregions recursively.) If such l' does not exist, then the strategy cannot be applied to R_j. In this case, the interior of R_j must be cleared only by using the searchlights located in $R_j - b(R_j)$.

Let us characterize the subregions found during recursive applications of the One-Way Sweep Strategy to which the strategy cannot be applied in step 2.2.

Definition 5 Let $P = (D, L)$ be an instance. A closed simple subregion $R \subseteq D$ is called a *maximal isolated region* if the following conditions hold.

1. For any searchlight $l \in L$, if $(R - b(R)) \cap V_l \neq \phi$, then $l \in R - b(R)$.
2. For any point $x \in b(R)$, either $x \in b(D)$ or $x \in V_l$ for some searchlight $l \in L$ such that $l \notin R - b(R)$.

The following lemma is straightforward, and the proof is thus omitted.

Lemma 1 *Let $P = (D, L)$ be an instance such that $b(D) \cap L \neq \phi$. Assume that the One-Way Sweep Strategy is applied recursively to D and its subregions. Let $R \subseteq D$ be a subregion found during the execution for which the strategy cannot be applied recursively at step 2.2. Then R is a maximal isolated region.*

Theorem 1 *Let $P = (D, L)$ be an instance such that $b(D) \cap L \neq \phi$. There exists a search schedule for P iff there exists a search schedule for the instance $P_R = (R, L_R)$ for every maximal isolated region $R \subseteq D$, where $L_R = (R - b(R)) \cap L$ is the set of searchlights located in $R - b(R)$.*

Proof (if) Apply the One-Way Sweep Strategy to D and its subregions recursively. Whenever a subregion $R \subseteq D$ to which the strategy cannot be applied is found in step 2.2, execute a search schedule for $P_R = (R, L_R)$. This is possible by Lemma 1 and the assumption of this theorem. Clearly, the resulting process is a search schedule for P.
(only if) Let F be a search schedule for P. Let $F_R = \{f_l \in F \mid l \in L_R\}$. Let r be an arbitrary trajectory function from real time to R which cannot be detected by the searchlights in L_R when F_R is executed. Since r is not detected, if we let δ be the minimum, over all t, of the distances between $r(t)$ and the rays of the searchlights of $l \in L_R$ at time t, then we have $\delta > 0$. This implies that there exists another trajectory function r' such that $r'(t) \in R - b(R)$ for any t which cannot be detected when F_R is executed. Since F is a search schedule for P, there exist some $l \in L$ and t such that l is aimed at $r'(t)$ at time t when F is executed. Here, since $r'(t) \in R - b(R)$, by condition 1 of Definition 5, we have $l \in L_R$. This is a contradiction. □

Since the interior of a maximal isolated region R must be cleared only by using the searchlights located in $R - b(R)$, by Theorem 1 we have reduced the problem of finding a search schedule for an instance $P = (D, L)$ having at least one searchlight on the boundary of D to that for instances having no searchlight on the boundary.

We conclude this section by introducing three sufficient conditions for the existence of a search schedule.

Definition 6 Let $P = (D, L)$ be an instance. The *searchlight visibility graph* of P is an undirected graph $SVG(P) = (L, E)$ with vertex set L and edge set E such that for any $l, l' \in L$, $(l, l') \in E$ iff $l \neq l'$ and $l \in V_{l'}$.

Definition 7 An instance $P = (D, L)$ is said to be *stable* if for every connected component $G_i = (L_i, E_i)$ of $SVG(P)$, there is at least one searchlight $l \in L_i$ such that $l \in b(D)$.

Theorem 2 *If $P = (D, L)$ is stable, then there exists a search schedule for P.*

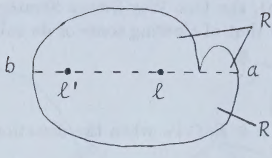


Figure 4: Illustration for Lemma 2.

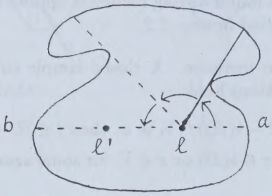


Figure 5: An additional searchlight l'.

Proof In view of Theorem 1, it suffices to show that there exists no maximal isolated region $R \subseteq D$ if P is stable. Suppose that there exists a maximal isolated region $R \subseteq D$. Let $P_R = (R, L_R)$, where $L_R = (R - b(R)) \cap L$ is the set of searchlights located in $R - b(R)$. By condition (P1) and condition 1 of Definition 5, (1) $L_R \neq \phi$, (2) any connected component of $SVG(P_R)$ is a connected component of $SVG(P)$, and (3) $L_i \cap b(D) = \phi$ for any connected component $G_i = (L_i, E_i)$ of $SVG(P_R)$. This contradicts the assumption that P is stable. □

We need the following definitions. Let R be any closed simple region. A line segment $\overline{ab}$ is called a *cut* of R if $a, b \in b(R)$ and $\overline{ab} \subseteq R$. A cut divides R into two or more subregions. A cut $\overline{ab}$ of R is *maximal* if $\overline{cd} \not\subseteq R$ for any points c and d such that $\overline{ab} \subset \overline{cd}$.

The following lemma is used in the proof of Theorem 3 and in later discussions.

Lemma 2 *Let $P = (D, L)$ be an instance. Assume that $l \in V_{l'}$ for some searchlights l and $l' \in L$. Let $\overline{ab}$ be the maximal cut such that a, l, l' and b appear in this order in $\overline{ab}$. Let R_1 and R_2 be the closed regions whose boundaries are $[a, b]_{b(D)} \cup \overline{ba}$ and $[b, a]_{b(D)} \cup \overline{ab}$, respectively (Figure 4). There exists a search schedule for P iff the following two conditions hold.*

1. *The following state is reachable from the initial state.*
   - *(a) l and l' are aimed at each other, and*
   - *(b) R_1 is clear.*
2. *The following state is reachable from the initial state.*
   - *(a) l and l' are aimed at each other, and*
   - *(b) R_2 is clear.*

Proof The necessity is obvious. We prove the sufficiency. Suppose that the state described in condition 1 has been reached. If R_2 is not clear at this state, then we execute the "reversal" of a schedule for realizing the state described in condition 2. Then D becomes clear. □

Theorem 3 *Let $P = (D, L)$ be an instance. If $SVG(P)$ is connected, then there exists a search schedule for the instance $P' = (D, L \cup \{l'\})$, where $l' \in D$ is an arbitrary searchlight not in L.*

Proof Since $D = \bigcup_{l \in L} V_l$ by condition (P1), there exists a searchlight $l \in L$ such that $l' \in V_l$. Let $\overline{ab}$ be the the maximal cut such that a, l, l' and b appear in this order in $\overline{ab}$ (Figure 5). Let D_u and D_l be the closed regions whose boundaries are $[a, b]_{b(D)} \cup \overline{ba}$ and $[b, a]_{b(D)} \cup \overline{ab}$, respectively. Suppose that we apply

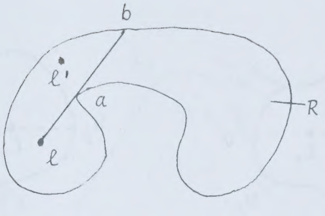


Figure 6: l is aimed at a and b.

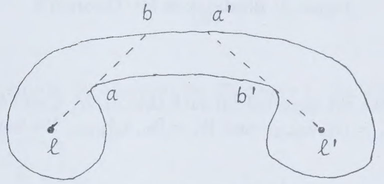


Figure 7: Illustration for Theorem 4.

the One-Way Sweep Strategy to D_u by keeping l' aimed at a and turning l counterclockwise from a to b. Since the connectedness of $SVG(P)$ assures that there exists no maximal isolated region $R \subseteq D_u$ since l is regarded as a searchlight on the boundary of D_u, by Theorem 1, when the execution of the strategy is completed l and l' are aimed at each other and D_u is clear. Note that the direction of l' need not be changed during the execution. Similarly, another state in which l and l' are aimed at each other and D_l is clear is reachable from the initial state. Thus by Lemma 2 there exists a search schedule for P'. □

The following lemma can be used to show the existence of a search schedule under certain conditions.

Lemma 3 *Let $P = (D, L)$ be an instance. For an arbitrary searchlight $l \in L$, let $(a, b)_{b(D)} \subseteq b(D) - V_l$ be a maximal open segment of $b(D)$ not visible from l, and let R be the closed simple region whose boundary is $[a, b]_{b(D)} \cup \overline{ba}$. If $SVG(P)$ is connected, then R can be cleared while l is kept aimed at a and b.*

Proof Suppose that l is aimed at a and b (Figure 6). By condition (P1) and the connectedness of $SVG(P)$, there exists a searchlight l' such that $l' \notin R - b(R)$ and $(R - b(R)) \cap V_{l'} \neq \phi$. Thus the One-Way Sweep Strategy can be applied to R. Since there exists no maximal isolated region $R' \subseteq R$ by the connectedness of $SVG(P)$, by Theorem 1 R becomes clear when the execution of the strategy is completed. □

Theorem 4 *Let $P = (D, L)$ be an instance. If $SVG(P)$ is connected and there exist two searchlights $l, l' \in L$ such that $V_l \cap V_{l'}$ is either ϕ or a single point, then there exists a search schedule for P.*

Proof Let $(a, b)_{b(D)} \subseteq b(D) - V_l$ be the maximal open segment of $b(D)$ not visible from l such that $l' \in R$, where R is the closed simple region whose boundary is $[a, b]_{b(D)} \cup \overline{ba}$. Similarly, let $(a', b')_{b(D)} \subseteq b(D) - V_{l'}$ be the maximal open segment of $b(D)$ not visible from l' such that $l \in R'$, where R' is the closed simple region whose boundary is $[a', b']_{b(D)} \cup \overline{b'a'}$ (Figure 7). Since $SVG(P)$ is connected, by Lemma 3 R can be cleared. At this state $D - R'$ is clear, since $V_l \cap V_{l'}$ is either ϕ or a single point. Again by Lemma 3, starting from this state R' can be cleared while l' is kept aimed at a' and b'. Thus D becomes clear. □

4 Instances Having No Searchlight on the Boundary

Let $P = (D, L)$ be an instance such that $b(D) \cap L = \phi$. We present various conditions for the existence of a search schedule for P when the number of searchlights $|L|$ is two or three. The proofs of the results presented in this section may be found in [5].

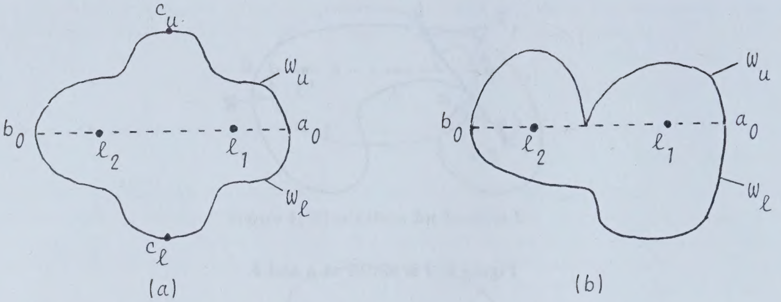


Figure 8: Illustrations for Theorem 5.

4.1 $|L| = 2$

Let $L = \{l_1, l_2\}$. Let $\overline{a_0 b_0}$ be the the maximal cut such that a_0, l_1, l_2 and b_0 appear in this order in $\overline{a_0 b_0}$, as shown in Figure 8. Let $W_u = [a_0, b_0]_{b(D)}$ and $W_l = [b_0, a_0]_{b(D)}$. We have the following necessary and sufficient condition.

Theorem 5 *There exists a search schedule for P iff one of the following conditions holds.*

1. *There exist points $c_u \in W_u$ and $c_l \in W_l$ such that $[c_u, c_l]_{b(D)} \in V_{l_1}$ and $[c_l, c_u]_{b(D)} \in V_{l_2}$ (Figure 8(a)).*
2. *For each $s \in \{u, l\}$, $W_s \cap \overline{l_1 l_2} \neq \phi$ or $W_s \subseteq V_{l_i}$ for some $i \in \{1, 2\}$ (Figure 8(b)).*

The reader can verify that the instance shown in Figure 2 satisfies neither condition 1 nor 2 of Theorem 5.

4.2 $|L| = 3$

Definition 8 $P = (D, L)$ is said to be *shallow* if for any searchlight $l \in L$ and any maximal open region $R \subseteq D - V_l$ not visible from l such that $R \cap L = \phi$, there exists a searchlight $l' \in L$ such that $R \subseteq V_{l'}$.

The instances shown in Figures 1 and 10 are both shallow.

The case of $|L| = 3$ turns out to be quite complex. Let $L = \{l_1, l_2, l_3\}$. Without loss of generality assume that l_3 lies to the left of the oriented line segment from l_1 to l_2. We have three cases.

4.2.1 $l_i \notin V_{l_j}$ for some $i, j \in \{1, 2, 3\}$

Without loss of generality assume that $l_1 \notin V_{l_2}$. We use the symbols given in Figure 9(a). The line segment $\overline{a_3 b_1}$ is the maximal cut such that a_3, l_3, l_1 and b_1 appear in this order in $\overline{a_3 b_1}$, and c_1 is the point in $\overline{l_1 b_2} \cap b(D)$ closest to l_1. Other symbols are defined similarly.

Theorem 6 *There exists a search schedule for P if P is shallow and one of the following conditions holds.*

(C1) $[a_3, b_3]_{b(D)} \subseteq V_{l_2} \cup V_{l_3}$.

(C2) $[a_3, b_3]_{b(D)} \subseteq V_{l_1} \cup V_{l_3}$.

(C3) $[b_1, c_1]_{b(D)} \subseteq V_{l_3}$.

(C4) $[c_2, a_2]_{b(D)} \subseteq V_{l_3}$.

(C5) $l_3 \in \overline{c_1 b_2}$.

(C6) $l_3 \in \overline{c_2 a_1}$.

(C7) *There exist points $p, q, r \in [b_2, a_1]_{b(D)}$ satisfying the following conditions simultaneously.*

1. *p, q and r appear in this order in $[b_2, a_1]_{b(D)}$.*
2. *$(p, r)_{b(D)} \subseteq b(D) - V_{l_1}$ is a maximal open segment not visible from l_1.*

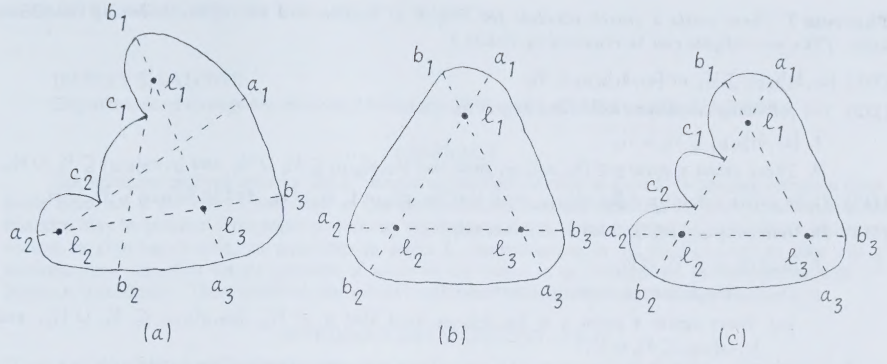


Figure 9: Illustrations for Theorems 6, 7 and 8.

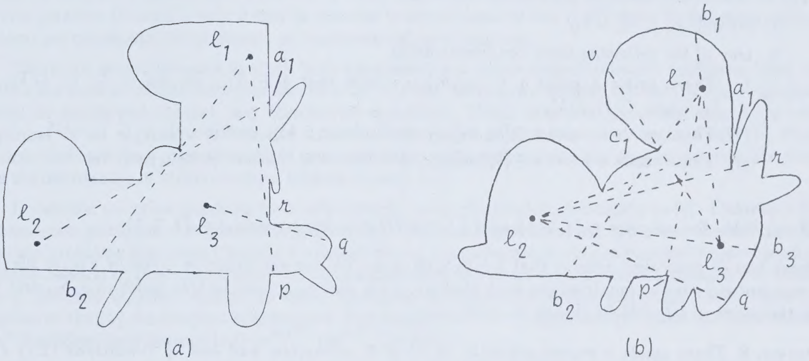


Figure 10: Instances satisfying (a) condition (C7) and (b) condition (C9).

3. $[b_2, p]_{b(D)} \subseteq V_{l_1}$, $[p, q]_{b(D)} \subseteq V_{l_3}$, $[q, r]_{b(D)} \subseteq V_{l_2}$, *and* $[r, a_1]_{b(D)} \subseteq V_{l_1} \cup V_{l_3}$.

(C8) *This is symmetric to (C7) and is thus omitted.*

(C9) *The following conditions hold simultaneously.*

1. *There exist points* $p, q, r \in [b_2, a_1]_{b(D)}$ *such that*
   - *(a)* p, q *and* r *appear in this order in* $[b_2, a_1]_{b(D)}$,
   - *(b)* $r \in [b_3, a_1]_{b(D)}$,
   - *(c)* l_2, p *and* q *are on a line, and*
   - *(d)* $[b_2, p]_{b(D)} \subseteq V_{l_1}$, $[p, q]_{b(D)} \subseteq V_{l_1} \cup V_{l_3}$, $[q, r]_{b(D)} \subseteq V_{l_2}$, *and* $[r, b_1]_{b(D)} \subseteq V_{l_3}$.
2. *The intersection of* $\overline{l_2 r}$ *and* $\overline{l_3 c_1}$ *lies to the left of the oriented line segment from* p *to* l_1.
3. *Let* $v \in [b_1, c_1]_{b(D)}$ *be the point closest to* c_1 *such that* $[b_1, v]_{b(D)} \subseteq V_{l_3}$. *Then* l_3, c_1 *and* v *are on a line.*

(C10) *This is symmetric to (C9) and is thus omitted.*

The reader can verify that the instance shown in Figure 1 satisfies condition (C1). The instances shown in Figure 10(a) and (b) satisfy conditions (C7) and (C9), respectively.

4.2.2 $\overline{l_i l_j} \cap b(D) = \phi$ for any $i, j \in \{1, 2, 3\}$

We use the symbols given in Figure 9(b). The line segment $\overline{a_1 b_2}$ is the maximal cut such that a_1, l_1, l_2 and b_2 appear in this order in $\overline{a_1 b_2}$. Other symbols are defined similarly.

Theorem 7 *There exists a search schedule for P if P is shallow and one of the following conditions holds. (The searchlights can be renamed cyclically.)*

(D1) $[a_1, b_1]_{b(D)} \subseteq V_{l_2}$ *or* $[a_1, b_1]_{b(D)} \subseteq V_{l_3}$.

(D2) *The following conditions hold simultaneously.*

1. $[a_1, b_1]_{b(D)} \subseteq V_{l_2} \cup V_{l_3}$.
2. *There exists a point* $p \in [b_2, a_3]_{b(D)}$ *such that* $[b_2, p]_{b(D)} \subseteq V_{l_1} \cup V_{l_3}$ *and* $[p, a_3]_{b(D)} \subseteq V_{l_1} \cup V_{l_2}$.

(D3) *There exists a point* $p \in [b_2, a_3]_{b(D)}$ *such that* $[a_2, p]_{b(D)} \subseteq V_{l_1} \cup V_{l_3}$ *and* $[p, b_3]_{b(D)} \subseteq V_{l_1} \cup V_{l_2}$.

(D4) *The following conditions hold simultaneously.*

1. $[b_3, b_1]_{b(D)} \subseteq V_{l_2} \cup V_{l_3}$.
2. *One of the following three conditions holds.*
    - *(a) There exists a point* $q \in [a_3, b_3]_{b(D)}$ *such that* $q \in V_{l_2}$, $[a_3, q]_{b(D)} \subseteq V_{l_1} \cup V_{l_2}$, *and* $[q, b_3]_{b(D)} \subseteq V_{l_2} \cup V_{l_3}$.
    - *(b) There exists a point* $r \in [a_2, b_2]_{b(D)}$ *such that* $r \in V_{l_3}$, *and* $[r, b_2]_{b(D)} \subseteq V_{l_1} \cup V_{l_3}$.
    - *(c) There exists a point* $s \in [b_2, a_3]_{b(D)}$ *such that* $s \in V_{l_3}$, *and* $[b_2, s]_{b(D)} \subseteq V_{l_1}$.

(D5) *The following conditions hold simultaneously.*

1. $[a_1, a_2]_{b(D)} \subseteq V_{l_2} \cup V_{l_3}$.
2. *One of the following three conditions holds.*
    - *(a) There exists a point* $q \in [a_2, b_2]_{b(D)}$ *such that* $q \in V_{l_3}$, $[a_2, q]_{b(D)} \subseteq V_{l_2} \cup V_{l_3}$, *and* $[q, b_2]_{b(D)} \subseteq V_{l_1} \cup V_{l_3}$.
    - *(b) There exists a point* $r \in [a_3, b_3]_{b(D)}$ *such that* $r \in V_{l_2}$, *and* $[a_3, r]_{b(D)} \subseteq V_{l_1} \cup V_{l_2}$.
    - *(c) There exists a point* $s \in [b_2, a_3]_{b(D)}$ *such that* $s \in V_{l_2}$, *and* $[s, a_3]_{b(D)} \subseteq V_{l_1}$.

4.2.3 $l_i \in V_{l_j}$ for any $i, j \in \{1, 2, 3\}$ and $\overline{l_i l_j} \cap b(D) \neq \phi$ for some $i, j \in \{1, 2, 3\}$

Without loss of generality assume that $\overline{l_1 l_2} \cap b(D) \neq \phi$. We use the symbols given in Figure 9(c). The line segment $\overline{a_1 b_2}$ is the maximal cut such that a_1, l_1, l_2 and b_2 appear in this order in $\overline{a_1 b_2}$, and c_1 (or c_2) is the point in $\overline{l_1 l_2} \cap b(D)$ closest to l_1 (or l_2).

Theorem 8 *There exists a search schedule for P if P is shallow and one of conditions (C1)-(C10), (D1)-(D5) holds.*

5 Conclusion

We have posed the Searchlight Scheduling problem and presented various conditions for the existence of a search schedule. Obtaining a simple necessary and sufficient condition for the existence of a search schedule for instances having three or more searchlights remains as a challenging open problem. The computational complexity of various questions remains to be investigated. The case in which a given region can contain holes seems to be even more challenging.

References

[1] V. Chvátal, "A combinatorial theorem in plane geometry," *J. Combinatorial Theory (B)* 18, 1975, pp. 39-41.

[2] J. Kahn, M. Klawe and D. Kleitman, "Traditional galleries require fewer watchmen," *SIAM J. Algebraic and Discrete Methods* 4, No. 2, 1983, pp. 194–206.

[3] D. T. Lee and A. K. Lin, "Computational complexity of Art Gallery problems," *IEEE Transactions on Information Theory* IT-32, No. 2, 1986, pp. 276–282.

[4] J. O'Rourke, *Art Gallery Theorems and Algorithms*, Oxford University Press, New York, 1987.

[5] K. Sugihara, I. Suzuki and M. Yamashita, "The searchlight scheduling problem," Tech. Rep., EECS Dept., University of Wisconsin–Milwaukee, in preparation.

GEOMETRIC SELECTION BY PARAMETRIC SEARCH

JEFFREY S. SALOWE
Department of Computer Science, University of Virginia, Charlottesville, VA 22903

ABSTRACT

The selection problem asks for the k^{th} largest or smallest element in a set S. In general, selection takes linear time, but if the set is constrained so that some relations between elements are known, sublinear-time selection may be possible. This paper presents an algorithm to select elements from a Cartesian sum of two n-vectors in $O(n \log n)$ time, an algorithm to select L_∞-interdistances in $\Re^d$ in $O(d\, n \log^d n)$ time and a sublinear-time algorithm for the problem of selecting the vertex in an arrangement of hyperplanes with k^{th} largest x_1-coordinate. These problems are solved using a new paradigm for selection algorithms.

INTRODUCTION AND MOTIVATION

Given a totally ordered set S of n elements, let $\rho_x = | \{y : y \le x\} |$ be the rank of element x. For inputs S and x, the ranking problem is to determine ρ_x, and for inputs S and integer k, $1 \le k \le |S|$, the selection problem is to return the element x for which $\rho_x = k$. The seminal paper of [4] proves that the k^{th} largest (or smallest) element in S can be selected in linear time. However, if some order relations between elements of S are known (i.e. the selection problem is constrained), it may be possible to select elements in $o(|S|)$ time. In this paper, selection problems are constrained by geometric or combinatorial considerations.

There are several reasons to study both constrained and unconstrained selection problems. First, order statistics are used to robustly estimate the behavior of a set. Second, the median is often used as a partitioning element in divide-and-conquer and elimination algorithms. Third, selection algorithms may be a crucial component in an algorithm for another task; an example of this phenomenon appears in [11]. Finally, sublinear-time selection algorithms have some information-theoretic value since they must make a statement about the distribution of objects without looking at every one.

Geometric selection problems have only recently received attention, principally in [6]. Chazelle's paper presented one paradigm for constructing geometric selection algorithms, but there are several interesting geometric problems that resist Chazelle's method; that is, his technique is not a substantial improvement over the trivial method of first enumerating the set and then using the linear time selection algorithm. For instance, there is no truly efficient method to select Euclidean interdistances in dimension $d \ge 2$, with the notable exception of finding the smallest interdistance. (For n points in $\Re^d$, the algorithm proposed by Chazelle selects the k^{th} Euclidean interdistance in $O(n^{2-2^{-(d+1)}} \log^{1-2^{-(d+1)}} n)$ time.)

We make the following additional contributions to the study of geometric selection problems. First, we describe a new technique (also used in [8]) for designing selection algorithms that is substantially different from that of Chazelle and which seems to be widely applicable to both geometric and combinatorial problems. We use this technique to design an $O(n \log n)$ time algorithm to select elements from a Cartesian sum, and we generalize this approach to yield an $O(d\, n \log^d n)$ time algorithm to select L_∞ interdistances in $\Re^d$. A pleasant side-effect of this algorithm is that it resolves an open problem similar to one suggested by Aggarwal [1]. Finally, we introduce the multidimensional analogue of the slope selection problem studied in [8], selecting vertices in an arrangement of hyperplanes, and we show how the parametric search technique can be used to produce a sublinear-time selection algorithm with respect to the number of vertices in the arrangement.

THE SELECTION TECHNIQUE AND AN EXAMPLE

Our approach to selection algorithms is based on a technique of using parallel algorithms in the design of efficient sequential algorithms [14], which in turn is a refinement of parametric search presented in [13]. Given a search problem **A**, a parametric search algorithm is one that finds a particular item s^* contained in search space S_A using another algorithm **B** to guide the search.

In fact, if problem **A** meets certain conditions and there is a good parallel algorithm for problem **B**, parametric search may be accelerated. The first step to do this is to simulate the parallel algorithm in a sequential fashion; that is, perform all the operations on the first level in any order (a level is a group of concurrent operations in the parallel algorithm), then the operations on the second level in any order, and so on.

Suppose there is an algorithm which uses some basic operation to guide the search for $s^* \in S_A$. Although each operation may be costly, operations may be "ordered" so that after performing one operation expensively, many others can be resolved cheaply. It is precisely in this circumstance that we can take advantage of the independence of operations in a parallel algorithm. On each level of the parallel algorithm, choose the operation that affects the greatest number of the remaining ones, partition the set of operations into those that are consequently resolved and those that are not, and then recurse on the set of operations that are not resolved. This has the effect of drastically reducing the number of expensive operations performed.

To demonstrate this technique, we present a new algorithm to select elements from a Cartesian sum. The Cartesian sum of m-vector X and n-vector Y is the set

$$X + Y = \{ x_i + y_j : 1 \le i \le m \ , \ 1 \le j \le n \}.$$

A theorem of Frederickson and Johnson [10] states that the k^{th} largest or smallest element in $X + Y$ can be selected in $O(m + p \log \frac{k}{p})$ time, $p = \min \{k,m\}$, which is optimal in the comparison tree model of computation. For the exposition, we assume that $m = n$ and that the elements in the Cartesian sum are distinct.

A merge algorithm is used to parametrically guide the search for the k^{th} largest element in $X + Y$, and the comparisons of the merge algorithm are resolved by ranking. We sketch how to rank in linear time after an $O(n \log n)$ time preprocessing step, and then we show how this ranking algorithm can be incorporated into an $O(n \log n)$ time selection algorithm.

First, sort X and Y, and suppose we are ranking an element in the Cartesian sum with value t. Consider merging the set X with the set $-Y + t = \{-y_i + t : y_i \in Y\}$. If a particular item in $-Y + t$, say $-y_i + t$, is greater than or equal to x_j, then $x_j + y_i \le t$. Therefore, the number of items in $X + Y$ whose value is less than or equal to t (i.e. the rank) can be determined in linear time by a left-to-right pass over the merge of X and $-Y + t$. The details are left to the reader.

Given the ability to rank, a selection algorithm can be designed using the parametric search technique. The idea is to merge the set X with the set $-Y + t^*$, where t^* is the value of the unknown k^{th} element in $X + Y$. Although this value is unknown, a comparison of the form "Is x_j less than $-y_i + t^*$?" can be resolved by ranking. If $\rho_{x_j + y_i} < k$, then $x_j + y_i < t^*$ and x_j is less than $-y_i + t^*$. If $\rho_{x_j + y_i} > k$, then $x_j + y_i > t^*$ and x_j is greater than $-y_i + t^*$. Otherwise, $\rho_{x_j + y_i} = k$, and $x_j + y_i$ is the k^{th} element in $X + Y$. Note that any merge algorithm must compare $-y_u + t^*$ with x_v, where $t^* = x_v + y_u$, since these two points are coincident in the merged order.

It is also essential that the comparisons have a desirable "ordering property." If one decides that $x_j + y_i$ is larger than t^*, then one knows in constant time that any $x_q + y_r > x_j + y_i$ is larger than t^*, and a similar statement can be made if $x_j + y_i$ is smaller than t^*. In this situation, it is useful to "batch" comparisons by level in a sequentialized version of a parallel algorithm. We use a sequentialized version of the parallel merge algorithm of [5, 18] to find the desired t^*. Valiant [18] showed that two lists of size n can be merged using $O(\log\log n)$ levels of $O(n)$ comparisons each, and these bounds also hold when overheads are counted [5]. The merge algorithm can be simulated sequentially in $O(n \log\log n)$ time.

To implement the accelerated parametric search of Megiddo [14], consider a particular level of the parallel algorithm. Rank the median comparison with respect to the "ordering property" mentioned above, resolve half of the comparisons on that level in constant time each, and continue in the same fashion with the unresolved comparisons. Of the $O(n)$ comparisons corresponding to a given level, $O(\log n)$ are resolved by a linear-time ranking and the remainder are resolved by transitivity. Since there are $O(\log\log n)$ levels, the cost of this algorithm is

$$O(n \log\log n) + O(n) \cdot O(\log n) \cdot O(\log\log n) = O(n \log n \log\log n).$$

In fact, using the strategy of Cole [7] of weighting comparisons based on their level and then determining the weighted median comparison over all levels, only $O(\log n)$ rankings need to be performed over the entire algorithm. The reader is referred to [7] for a complete explanation. Since each ranking can be done in linear time after the preprocessing step, the total cost of selection is $O(n \log n)$.

In summary, we have proved:

Theorem 1: The parametric search algorithm selects the k^{th} largest (or smallest) element in $X + Y$ in $O(n \log n)$ time.

The following geometric selection result is a corollary to the preceding theorem. Given any set X of n points in $\Re$, the set of one-dimensional interdistances is

$$\{ |x_i - x_j| : i < j \}.$$

It was shown in [17] that the median one-dimensional interdistance may be found in $O(n \log n)$ time since, assuming the x_i are distinct, the one-dimensional interdistances can be interpreted as the positive-valued elements in $X + -X$.

Corollary 1: The parametric search technique selects the k^{th} largest (or smallest) one-dimensional interdistance in $O(n \log n)$ time.

SELECTING L_∞-INTERDISTANCES

The L_∞ distance between two points x_i and x_j is defined as

$$\|x_i - x_j\|_\infty = \max_{1 \le k \le d} |x_{i_k} - x_{j_k}|.$$

In this section, we describe an algorithm that, given n points in $\Re^d$, finds the k^{th} L_∞-interdistance in $O(d\, n \log^d n)$ time. This appears to be the first algorithm for a multidimensional interdistance selection problem whose time complexity is in $O(n \log^{O(1)} n)$.

The known bounds on L_∞-interdistance selection are the following. (1) The lower bounds for the one-dimensional problem also hold for L_∞ since any one-dimensional problem can be embedded into an L_∞ problem of arbitrary dimension. Therefore, the smallest L_∞-interdistance and the median L_∞-interdistance require $\Omega(n \log n)$ time. (2) The smallest L_∞-interdistance can be computed in asymptotically optimal $O(n \log n)$ time using the divide-and-conquer algorithm of [3]. (3) The largest interdistance can be found in $O(dn)$ time by determining the maximum difference between the points in each of the coordinates, and (4) no $o(n^2)$ time upper bounds for the general selection problem are explicitly stated in the literature, but it is possible to extend Chazelle's L_2-interdistance algorithm to select L_∞-interdistances in $O(n^{2-2^{-(d+1)}} \log^{1-2^{-(d+1)}} n)$ time.

Let $X = \{x_1, \ldots, x_n\}$ be a set of n points in $\Re^d$, and assume that the set of L_∞-interdistances determined by X are distinct. Removing this assumption does not adversely affect the asymptotic complexity of the algorithm. Define $C^d(x_i, t)$ to be a closed d-cube of side $2t$ centered at input point x_i and aligned with the coordinate axes.

Theorem 2: Given a set X of n points in $\Re^d$, the k^{th} largest (or smallest) L_∞-interdistance can be selected in $O(d\, n \log^d n)$ time.

Proof: There are two parts to the proof. The first part is to determine the complexity of ranking and the second part is to organize the parametric search. The principal observation for ranking L_∞-interdistances is that the points within distance t of $x_i \in \Re^d$ are contained in $C^d(x_i, t)$. The problem of counting the number of points within an arbitrary d-cube is a special case of the orthogonal range counting problem for which Willard [19] devised the layered range tree with optimal space and query time (see [15]). With this structure, orthogonal range counting of n points in $\Re^d$, $d \ge 2$, can be effected by an algorithm which uses $O(n \log^{d-1} n)$ preprocessing time, $O(n \log^{d-1} n)$ space and $O(log^{d-1} n)$ time queries.

To rank distance t (not necessarily an interdistance), observe that for each $x_i \in X$, the points satisfying $\|x_i - x_j\|_\infty \le t$, $j \ne i$, are inside $C^d(x_i, t)$. The rank of t is determined by summing the orthogonal range counts and then dividing the result by 2 to compensate for double-counting; as a result, if X is a set of n points in $\Re^d$, $d \ge 2$, distance t can be ranked in time $O(n \log^{d-1} n)$.

The selection algorithm is parametrically driven by a collection of d one-dimensional merging problems whose comparisons are resolved by ranking. In the s^{th} such problem, input points $x_i \in X$ are projected onto the s^{th} coordinate axis by setting all but the s^{th} coordinate, x_{i_s}, to zero; these projected points are then sorted. For distance t, the projection of the boundary of $C^d(x_i, t)$ are the points $x_{i_s} - t$ and $x_{i_s} + t$. We are interested in determining the ordering of left endpoints and input points in each of the projected one-dimensional problems for unknown k^{th} interdistance t^*.

Cole's technique is used to choose a set of $O(d \log n)$ L_∞-distances (not necessarily interdistances) to rank. In the s^{th} merging subproblem, a comparison of $x_{i_s} - t^*$ with x_{j_s}, $x_{j_s} < x_{i_s}$, asks "Is $x_{i_s} - t^*$ less than x_{j_s}?" and is resolved by ranking $t_{ij_s} = |x_{i_s} - x_{j_s}|$. If $\rho_{t_{ij_s}} > k$, then $t_{ij_s} > t^*$ and $x_{i_s} - t^* < x_{j_s}$; if $\rho_{t_{ij_s}} < k$, then $t_{ij_s} < r^*$ and $x_{i_s} - t^* > x_{j_s}$, and if $\rho_{t_{ij_s}} = k$, then $t_{ij_s} = t^*$. It is clear that any pair of comparisons can be ordered by the distances they correspond to. Furthermore, in at least one of the projected subproblems, the k^{th} interdistance t^* is realized, and it must be ranked since the points determining the k^{th} interdistance are coincident in this subproblem.

Since the ranking procedure takes $O(n \log^{d-1} n)$ time and the selection algorithm makes $O(d \log n)$ rankings, a total of $O(d\, n \log^d n)$ time is needed. □

There are two interesting corollaries to this theorem. In 2-dimensions, L_1- and L_∞-interdistances are related by a rotation of $\frac{\pi}{2}$, so the following is immediately true.

Corollary 2: In 2-dimensions, the k^{th} largest (or smallest) L_1-interdistance can be selected in $O(n \log^2 n)$ time.

The second corollary answers a question similar to one raised by A. Aggarwal [1]. Let A and B be sets of n points in $\Re^3$, and define the set of $A-B$-interdistances to be

$$\{ \|x-y\|_2 : x \in A , y \in B \}.$$

Aggarwal asked whether it is possible to find the smallest $A-B$-interdistance in $O(n \log^{O(1)} n)$ time. It is straightforward to prove that the algorithm for L_∞-interdistance selection can be adapted to give an $O(n \log^3 n)$ time algorithm for the analogue of Aggarwal's problem for the L_∞ metric.

Corollary 3: Given two sets A and B in $\Re^d$, the k^{th} largest (or smallest) $A-B$-interdistance with respect to the L_∞ metric can be selected in $O(d\, n \log^d n)$ time. In particular, the smallest $A-B$-interdistance with respect to the L_∞ metric can be found in $O(d\, n \log^d n)$ time.

We summarize the known results for selecting median interdistances in Figure 1; the time bounds proved in this paper are marked with an asterisk.

SELECTING VERTICES IN ARRANGEMENTS OF HYPERPLANES

The second application of the parametric search technique concerns arrangements of hyperplanes. It has been found that many problems dealing with sets of points in Euclidean space are most easily solved as problems for arrangements. A great deal of attention has focused on discovering combinatorial properties of arrangements. We show that the questions "Given a set of hyperplanes H in $\Re^d$ and a vertex v in the corresponding arrangement, how many vertices in the arrangement are to the left of v?" and "Given a set of hyperplanes H in $\Re^d$, what is the vertex in the corresponding arrangement with median x_1-coordinate?" can be answered in sublinear time with respect to the number of vertices.

We define the following terminology with respect to sets of points in $\Re^d$. A p-flat is an affine subspace of dimension p. A 0-flat is called a point, a 1-flat is called a line, and a $(d-1)$-flat is called a hyperplane. Let $H = \{h_1, \ldots, h_n\}$ be a set of n hyperplanes. H induces a cell complex of $O(n^d)$ cells in $\Re^d$ called a d-arrangement, and this arrangement is denoted by $A_d(H)$. Each cell is convex, and the dimension of a cell is the

Interdistance Selection Results

norm (p)	dim (d)	Bound (median)	Proven Optimal?	Reference
-	1	$O(n \log n)$	Yes	[10, 17], *
1	2	$O(n \log^2 n)$	No	*
1	$d>2$	$o(n^2)$	No	[16]
2	$d \geq 2$	$O(n^{2-2^{-(d+1)}} \log^{1-2^{-(d+1)}} n)$	No	[6]
∞	$d \geq 2$	$O(d\, n \log^d n)$	No	*

Figure 1 — General Interdistance Selection Results

dimension of its affine hull. The cells in $A_d(H)$ of dimension 0 are called vertices and the cells of dimension 1 are called edges. Note that there are $O(n^d)$ vertices in $A_d(H)$.

The complete skeleton of arrangement $A_d(H)$ is the set of all its vertices and edges, but we shorten this name to the skeleton. By a theorem of Edelsbrunner, O'Rourke and Seidel [9], an incidence graph representing the skeleton of an arrangement can be constructed in optimal $O(n^d)$ time. From this graph, it is possible to list the vertex-edge sequence along any line of the skeleton by following pointers. Reference [9] provides a detailed description of this representation.

We assume that $A_d(H)$ is simple, which means that the common intersection of p hyperplanes in H, $p \le d$, is a $(d-p)$-flat, and we assume that no two vertices have the same x_1-coordinate. These assumptions make the following arguments easier without affecting the asymptotic time complexity of the algorithms.

The first step in the selection algorithm is the ranking algorithm. There are two distinct subproblems in the ranking computation. First, the vertex with the smallest x_1-coordinate (the leftmost vertex) must be found to establish a starting skeleton. Second, there must be a procedure which compares the intersection skeletons at t_1 and t_2 and returns the number of vertices $v_i = (x_{i_1}, \ldots, x_{i_d})$ for which $t_1 \le x_{i_1} \le t_2$.

Lemma 1: Given a set of n hyperplanes H, the leftmost vertex in $A_d(H)$ can be found in time $O(n^{d-1} \log n)$.

Proof: We describe a recursive algorithm for this task. As a basis, let P be a plane in $\Re^d$, $d \ge 2$, with unit normal e, and let $A_2(H \cap P)$ be a 2-arrangement of n lines in P. Also, let e_i be the unit normal corresponding to the i^{th} coordinate axis. If $e = \pm e_1$, any vertex in $A_2(H \cap P)$ is a leftmost vertex. Otherwise, orthogonally project $A_2(H \cap P)$ onto the $x_1 x_i$ plane, where $i > 1$ is the smallest index with $e \ne \alpha e_1 + \beta e_i$. Call this projected arrangement $A'_2(H \cap P)$. The leftmost vertex in $A'_2(H \cap P)$ may be found in $O(n \log n)$ time by sorting the projections of the lines in $H \cap P$ by slope, determining the intersection points of adjacent lines with respect to this sorted order and choosing the leftmost intersection point. Since the direction of projection is orthogonal to the x_1-axis, the leftmost vertex in $A'_2(H \cap P)$ corresponds to the leftmost vertex in $A_2(H \cap P)$.

Assume inductively that the leftmost vertex in any $(d-1)$-arrangement of n $(d-2)$-flats in $\Re^d$ lying on an arbitrary hyperplane can be found in $O(n^{d-2} \log n)$ time. Arrangement $A_d(H)$ can be decomposed into a collection of n $(d-1)$-arrangements of $(n-1)$ $(d-2)$-flats, each of which is denoted by $A^i_{d-1}(H - \{h_i\}) = A_d(H) \cap h_i$. Each p-face in $A_d(H)$, $p \le d-1$, is in $\bigcup_i A^i_{d-1}(H - \{h_i\})$, and a leftmost vertex must be in exactly d of the $A^i_{d-1}(H - \{h_i\})$ (this is true because of the simplicity assumption and the assumption that no two vertices have the same x_1-coordinate). To find the leftmost vertex in $A_d(H)$, choose the vertex with minimum x_1-coordinate in each of the $A^i_{d-1}(H - \{h_i\})$ and pick the minimum of this set. This procedure can be effected in $O(n \cdot (n-1)^{d-2} \log n + n^2) = O(n^{d-1} \log n)$ time. □

Theorem 3: Given a set of n hyperplanes H and a vertex $v \in A_d(H)$, the number of vertices to the left of v can be found in $O(n^{d-1} \log n)$ time.

Proof: The ranking procedure is based on inversion counts. Suppose hyperplane $x_1 = t$ sweeps past a vertex v_j of $A_d(H)$ with x_1-coordinate t_j. Let $v_j = h_{j_1} \cap \cdots \cap h_{j_d}$, and consider an arbitrary $(d-2)$-sized subset of these hyperplanes, $\{h_{j_1}, \ldots, h_{j_{d-2}}\}$. Examine the order in which the rest of the hyperplanes in H intersect line $h_{j_1} \cap \cdots \cap h_{j_{d-2}} \cap \{x_1 = t_j - \varepsilon\}$ of $A_{d-1}(H)[t_j - \varepsilon]$, $\varepsilon > 0$. There are two such sequences, one the reverse of the other. We choose the one in which $h_{j_{d-1}}$ precedes h_{j_d}, and we require that the orientation be consistent. As the hyperplane $x_1 = t$ sweeps further to the right, $h_{j_{d-1}}$ and h_{j_d} coincide in $A_{d-1}(H)[t_j]$, and $h_{j_{d-1}}$ follows h_{j_d} in $A_{d-1}(H)[t_j + \varepsilon]$, $\varepsilon > 0$. In the ordered list for $\{h_{j_1}, \ldots, h_{j_{d-2}}\}$, there is one additional inversion with respect to the starting sequence. Likewise, there is one additional inversion in the lists for each of the $\binom{d}{2}$ $(d-2)$-sized subsets of $\{h_{j_1}, \ldots, h_{j_d}\}$. At each vertex v_j, the total number of inversions along the lines of $A_{d-1}(H)[t]$ is increased by exactly $\binom{d}{2}$, one for each line incident with v_j.

To implement this procedure, construct the starting intersection skeleton, $A_{d-1}(H)[t]$, for an appropriate t to the left of the vertex in $A_d(H)$ with smallest x_1-coordinate. By Lemma 1, the leftmost vertex can be found in $O(n^{d-1} \log n)$ time. For each line in the intersection skeleton, the initial list of vertices is computed by following pointers in the incidence graph representation. Then, given an input vertex $v_i = (x_{i_1}, \ldots, x_{i_d})$, construct $A_{d-1}(H)[x_{i_1} + \varepsilon]$ and compute the ordered list of vertices along the lines determined by each $(d-2)$-set of hyperplanes. (The term ε disambiguates the order of the lists at a vertex.) This takes $O(n^{d-1})$ time and space, and for each line we can calculate the number of inversions from the starting list in $O(n \log n)$ time [12]. The rank is this total number of inversions divided by $\binom{d}{2}$. Since there are $O(n^{d-2})$ lists, the overall time complexity of ranking is $O(n^{d-1} \log n)$. □

Theorem 4: Given a set H of n hyperplanes in $\Re^d$, the vertex in $A_d(H)$ with k^{th} largest (or smallest) x_1-coordinate can be selected in $O(d\, n^{d-1} \log^2 n)$ time.

Proof: The ranking procedure can be incorporated into the parametric search. In this case, the guiding algorithm is sorting, and the sorting network of [2] (the AKS network) is used. For an input of size n, the AKS sorting network consists of $O(\log n)$ levels of $O(n)$ comparisons each. The network can be simulated sequentially in $O(n \log n)$ time.

The sorting algorithm must determine the $\begin{bmatrix} n \\ d-2 \end{bmatrix}$ orderings of intercepts along the lines in the intersection skeleton at unknown k^{th} vertex $v^* = (x_1^*, \ldots, x_d^*)$. Suppose $h_{i_{d-1}} < h_{i_d}$ in the initial ordering for $\{h_{i_1}, \ldots, h_{i_{d-2}}\}$. Each comparison is of the form, "Is $h_{i_{d-1}} < h_{i_d}$ in the list for $\{h_{i_1}, \ldots, h_{i_{d-2}}\}$ in $A_{d-1}(H)[x_i^*]$?" and can be resolved by ranking the vertex $v_{test} = h_{i_1} \cap \cdots \cap h_{i_d}$. Let $x_1(v)$ be the x_1-coordinate of v, and note that the number of vertices to the left of the sweep hyperplane is monotonically nondecreasing with respect to x_1-coordinate. Therefore, if $\rho_{v_{test}} < k$, $x_1(v^*) > x_1(v_{test})$, and $h_{i_{d-1}} > h_{i_d}$ in the list for $\{h_{i_1}, \ldots, h_{i_{d-2}}\}$ in $A_{d-1}(H)[x_1^*]$. Likewise, if $\rho_{v_{test}} > k$, then $x_1(v^*) < x_1(v_{test})$, and $h_{i_{d-1}} < h_{i_d}$ in the list for $\{h_{i_1}, \ldots, h_{i_{d-2}}\}$ in $A_{d-1}(H)[x_1^*]$. Otherwise, $\rho_{v_{test}} = k$ and $v^* = v_{test}$.

To obtain these orderings, $\begin{bmatrix} n \\ d-2 \end{bmatrix}$ sorting networks are run simultaneously, one for each line in the intersection. These networks can be viewed as a single network with $O(\log n)$ levels and $O(n^{d-1})$ comparisons per level. The result of [7] shows that $O(d \log n)$ rankings suffice to resolve all $O(n^{d-1} \log n)$ comparisons, so the selection algorithm uses $O(d\, n^{d-1} \log^2 n)$ time overall. ☐

In fact, the time complexity of selection can be reduced by a small factor using an approximation scheme due to Cole, Salowe, Steiger and Szemeredi. We first describe the approximation for $d = 2$, and then present the multidimensional analogue.

In two dimensions, the starting intersection skeleton consists of n points along the vertical line $x_1 = t$, where t is to the left of the vertex in $A_2(H)$ with smallest x_1-coordinate. If we assume the lines in the arrangement are indexed in order of increasing slope, the indices are initially sorted. For any other intersection skeleton, the top-to-bottom order of indices is permuted, and the number of inversions in this permutation is precisely the rank of the corresponding vertex. Our goal is to have an efficient, accurate approximation to the number of inversions in the intersection skeleton.

The top level of the approximation algorithm estimates the number of inversions between the intercepts along reference line r and the intercepts along the vertical query line q incident to v_{test}. A reference line has the property that its rank is known exactly; the approximate rank of vertex v_{test} is thus the sum or difference of the number of inversions at the reference line r and the number of inversions between r and the query line q. The absolute error e of the latter approximation depends on the true number of inversions between r and q.

Denote the approximate number of inversions between r and q by K', let K be the true number of inversions, let K^* be the number of inversions between r and the vertical line incident to v^*, and suppose that both q and v^* are to the right of r. By definition, $K \in (K' - e, K' + e)$. Whenever $K^* \notin (K' - e, K' + e)$, the relative order of v_{test} and v^* can be determined by the approximation. If $K^* \in (K' - e, K' + e)$, the ordering of v_{test} and v^* cannot be deduced from the approximation. In this case, $\rho_{v_{test}}$ is computed exactly and q is made the new reference line.

The actual approximation consists of the following steps. Assume that the intercepts along r are numbered in top-to-bottom sequence from 1 to n; the initial r is the line passing through the leftmost vertex. If $G \subseteq [n]$, let the image of the i^{th} point in G at q be $\Pi_G(i)$. The estimate will sort a small subset of $\Pi_{[n]}$ and use the detected number of inversions to calculate the estimate for K.

The first step is to divide the points at r into $\left\lfloor \frac{n}{\tau} \right\rfloor$ contiguous intervals of size $\lceil \tau \rceil$ and possibly one group of size less than $\lceil \tau \rceil$. The proof of correctness shows that an appropriate τ is $\tau = 2^{\sqrt{\log n}}$. For each interval G_j, merge-sort $\Pi_{G_j}(i)$, $1 \le i \le |G_j|$. This has the effect of ordering the image of G_j as well as counting the number of inversions between elements of G_j. This number of inversions is added to K'.

The elements $\Pi_{G_j}(i)$ are now sorted. Assign the first $\left\lceil \tau^{3/4 + \varepsilon/2} \right\rceil$ elements and the last $\left\lceil \tau^{3/4 + \varepsilon/2} \right\rceil$ elements weight 1. A small sample of the remaining elements is taken. Divide the remaining elements into intervals of length $\gamma = \left\lceil \tau^{1/2} \right\rceil$, assign the median of each interval weight γ, and assign the rest weight zero. From the topmost interval G_1 to the bottommost interval, place the elements of positive weight, in their order at q, into a list l. The elements within a group are ordered with respect to q, but elements in different groups are ordered with respect to r.

The estimate K' to the number of inversions between r and q is the number of inversions within the G_m's plus the estimated number of inversions between the G_m's. The number of inversions within the G_m's was counted when the groups were sorted, and the estimated number of inversions between the G_m's may be counted by an algorithm that contributes $weight(i) \cdot weight(j)$ to K' if i and j invert (i is above j at r but j is above i at q). Such an algorithm can be constructed from a merge-sort procedure and uses $O(|I| \log |I|)$ time.

We now prove that the approximation scheme is correct. Specifically, we show that the approximation takes $O(n^{d-1} \log^{1/2} n)$ time for the d-dimensional problem, and the approximation fails to distinguish the order of v^* and v_{test} at most $O(\sqrt{\log n})$ times. A full ranking is needed if the approximation fails to distinguish the order of v^* and v_{test}.

Recall that K is the actual number of inversions between r and q, K' is the approximate number of inversions, and e is the absolute error. We show that $e \le \frac{2K'}{\lfloor \tau^{\varepsilon} \rfloor}$, $0 < \varepsilon < \frac{1}{2}$, and then explain why this behavior is useful.

The bound on the error is obtained by comparing the estimated number of inversions between two groups G_i and G_j, $i < j$, with the true number of inversions. Let K_{ST} be the number of times an element from S crosses an element from T, and let K'_{ST} and e_{ST} be the restrictions of K' and e to sets S and T. Assume that ε is a constant, $0 < \varepsilon < \frac{1}{2}$. We show that for every pair G_i and G_j, $i \neq j$,

$$K_{G_iG_j} \in \left(K'_{G_iG_j} - \frac{2K'_{G_iG_j}}{\lfloor \tau^{\varepsilon} \rfloor},\ K'_{G_iG_j} + \frac{2K'_{G_iG_j}}{\lfloor \tau^{\varepsilon} \rfloor}\right)$$

which implies that

$$K \in \left(K' - \frac{2K'}{\lfloor \tau^{\varepsilon} \rfloor},\ K' + \frac{2K'}{\lfloor \tau^{\varepsilon} \rfloor}\right).$$

The proof is divided into cases based on the weights of the elements.

Lemma 2: Let $x \in G_j$, $weight(x) = 1$. Then either $K'_{G_ix} = K_{G_ix}$ or $K'_{G_ix} \ge \lceil \tau^{3/4 + \varepsilon/2} \rceil$ and $e_{G_ix} \le \frac{\gamma}{2}$.

Proof: If x crosses all of G_i, $K'_{G_ix} = K_{G_ix} = \lceil \tau \rceil$, while if x does not cross any element in G_i of weight 1, $K'_{G_ix} = K_{G_ix} = 0$. Otherwise, the image of x lies between the images of z_1 and z_2, with $z_1, z_2 \in G_i$ and z_1 above z_2. If z_1 and z_2 have weight 1, then $K'_{G_ix} = K_{G_ix}$. If z_1 has weight 1 and z_2 has weight γ, then K'_{G_ix} might overestimate K_{G_ix} by $\frac{\gamma}{2}$. If z_1 has weight γ and z_2 has weight 1, K'_{G_ix} might underestimate K_{G_ix} by $\frac{\gamma}{2}$. Otherwise, $weight(z_1) = weight(z_2) = \gamma$. In this case, the error can vary from $-\frac{\gamma}{2}$ to $\frac{\gamma}{2}$. In each of the latter three cases, element x has crossed the lower $\lceil \tau^{3/4 + \varepsilon/2} \rceil$ elements of G_i, each of which has weight 1 and is counted in the approximation. □

Denote the set of elements in G_i of weight γ by g_i. A bound on the error contributed by these elements is given in the following lemma.

Lemma 3:

$$e_{g_ig_j} \le \left\lceil \tau^{\frac{3}{2}} \right\rceil.$$

Proof: Suppose that G_i is above G_j at r, and consider the ordered sequence of $g_i \cup g_j$ at q. Denote this as

$$\cdots x_{i_1} \cdots x_{i_a} y_{j_1} \cdots y_{j_b} x_{i_{a+1}} \cdots x_{i_c} y_{j_{b+1}} \cdots$$

where $x_{i_m} \in g_i$ and $y_{j_m} \in g_j$. Each x or y is the median of a small group of size γ. Let p be the number of alternations from y to x in this sequence and note that $0 \le p \le \gamma$.

Each x_{i_a} corresponds to an interval in G_i of size γ, but errors may occur due to the coarseness of the approximation. If x_{i_a} inverts with adjacent y_{j_m}, then the approximation adds $\lceil \tau \rceil$ to the sum, though as few as $\frac{\lceil \tau \rceil}{4}$ inversions must occur in the exact count. Similarly, if x_{i_a} does not cross y_{j_m}, the approximation counts 0 inversions, though as many as $\frac{\lceil \tau \rceil}{4}$ may occur in the exact count.

The worst overestimation happens in the following situation. (The worst underestimation is similar.) At a $y_1 \cdots y_m \to x_1 \cdots x_s$ block, $\frac{\gamma}{2} - 1$ of the elements in the group represented by y_m do not cross the block $x_1 \cdots x_s$. Similarly, $\frac{\gamma}{2} - 1$ of the elements in the group represented by x_1 do not cross the block $y_1 \cdots y_m$. If we neglect insignificant terms, it is not hard to show that the error is at most $(s + m)\frac{\lceil \tau \rceil}{2} - \frac{\lceil \tau \rceil}{4}$. Summing this error over all the possible interfaces between x's and y's proves the lemma. □

Lemma 4:

$$K_{G_iG_j} \in \left(K'_{G_iG_j} - \frac{2K'_{G_iG_j}}{\lfloor \tau^{\varepsilon} \rfloor},\ K'_{G_iG_j} + \frac{2K'_{G_iG_j}}{\lfloor \tau^{\varepsilon} \rfloor}\right)$$

for $0 < \varepsilon < \frac{1}{2}$.

Proof: Assume that G_i is above G_j at r. For the proof, abbreviate $K_{G_iG_j}$ by K, and do the same for K' and e. We show that $e \le \frac{2K'}{\lfloor \tau^{\varepsilon} \rfloor}$; there are two cases.

Case 1: No weight γ items cross. Then the only intersections which contribute to the estimation are those between weight 1 elements or those between weight 1 elements and weight γ elements. Denote the lowest $\lceil \tau^{3/4+\varepsilon/2} \rceil$ elements in G_i by L_{G_i}, and denote the highest $\lceil \tau^{3/4+\varepsilon/2} \rceil$ elements in G_j by U_{G_j}. In $\min(U_{G_j})$ is below some element of L_{G_i}, then the only intersections and errors possible are those determined by Lemma 2. Therefore for each element x in G_j, either $e_{G_ix} = 0$ or

$$e_{G_ix} \le \frac{\gamma}{2} \le \left\lceil \tau^{3/4-\varepsilon/2} \right\rceil \le \frac{K'_{G_ix}}{\lfloor \tau^{\varepsilon} \rfloor}.$$

If $\min(U_{G_j})$ crosses $\max(L_{G_i})$, then there may be an additional error of $(\frac{\gamma}{2} - 1)^2$ due to weight 0 items between the weight γ items. Nevertheless,

$$e < \left(\frac{\gamma}{2} - 1\right)^2 + \gamma \cdot \left\lceil \tau^{3/4+\varepsilon/2} \right\rceil < \frac{\lceil \tau \rceil}{4} + \left\lceil \tau^{5/4+\varepsilon/2} \right\rceil \le \left\lceil \tau^{3/2} \right\rceil \le \frac{\left\lceil \tau^{3/2+\varepsilon} \right\rceil}{\lfloor \tau^{\varepsilon} \rfloor} \le \frac{K'}{\lfloor \tau^{\varepsilon} \rfloor}.$$

Case 2: Some weight γ items cross. Then Lemmas 2 and 3 show that the error is bounded by

$$\left\lceil \tau^{3/2} \right\rceil + 2\left\lceil \tau^{3/4+\varepsilon/2} \right\rceil \cdot \frac{\gamma}{2} < 2\left\lceil \tau^{3/2} \right\rceil.$$

The first term is due to weight γ elements and the second term is due to weight 1 elements. However, since all of U_{G_j} crosses all of L_{G_i},

$$e < 2\left\lceil \tau^{3/2} \right\rceil \le \frac{2\left\lceil \tau^{3/2+\varepsilon} \right\rceil}{\lfloor \tau^{\varepsilon} \rfloor} \le \frac{2K'}{\lfloor \tau^{\varepsilon} \rfloor}.$$

□

As a consequence of the error bound, the following lemma gives a bound on the number of reference line updates.

Lemma 5: Let $K \in (K' - \frac{2K'}{\lfloor \tau^{\varepsilon} \rfloor}, K' + \frac{2K'}{\lfloor \tau^{\varepsilon} \rfloor})$. Then at most $O(\frac{\log n}{\varepsilon \log \tau})$ reference line updates are needed.

Proof: Let K^*_j be the actual number of inversions between the j^{th} reference line and the vertical line passing through v^*, and let K'_j be the approximation to the number of inversions between the j^{th} reference line and q. Recall that an update is needed whenever $K^* \in (K' - \frac{2K'}{\lfloor \tau^{\varepsilon} \rfloor}, K' + \frac{2K'}{\lfloor \tau^{\varepsilon} \rfloor})$, so the maximum number of inversions between the vertical line incident with v^* and q is $\frac{4K'}{\lfloor \tau^\varepsilon \rfloor}$. Since q is made the new reference line, it is evident that

$$K_i^* \le \frac{4}{\lfloor \tau^\varepsilon \rfloor} K'_{i-1} \le \frac{4}{\lfloor \tau^\varepsilon \rfloor} K_{i-1}^* \left(1 - \frac{2}{\lfloor \tau^\varepsilon \rfloor}\right)^{-1}$$

where the latter inequality is true because a reference line update was needed. Since $K_0^* \le \binom{n}{2}$, it can be shown that the smallest number of iterations to ensure that $K_i^* \le 1$ is

$$i = O\left(\frac{\log n}{\varepsilon \log \tau}\right).$$

☐

Theorem 5: Given a 2-arrangement $A_2(H)$, the vertex with k^{th} largest (or smallest) x_1-coordinate can be selected in $O(n \log^{3/2} n)$ time.

Proof: The cost of the approximation is made up of the following terms. First $O(\tau \log \tau)$ time is needed to sort each G_i and determine its internal inversions. For the $\left\lfloor \frac{n}{\tau} \right\rfloor + 1$ groups, the cost is $O(n \log \tau)$ time. Second, given this sorting information the list l is formed in $O(\frac{n}{\tau} \cdot \tau^{3/4+\varepsilon/2})$ time, and calculating weighted inversions takes $O(|l| \log |l|)$ time. The cost of the approximation is $O(\frac{n \log n}{\tau^{1/4-\varepsilon/2}} + n \log \tau)$, and at most $O(\log n)$ approximations are made.

By Lemma 6, at most $O(\frac{\log n}{\varepsilon \log \tau})$ full rankings are made at a cost of $O(n \log n)$ per ranking. There is also an $O(n \log n)$ term to simulate the AKS network. If $\tau = 2^{\sqrt{\log n}}$, the overall time complexity is $O(n \log^{3/2} n)$.
☐

It is easy to extend this approximation to the d-dimensional problem. Let l be an arbitrary line in the intersection skeleton of $A_{d-1}(H)[t]$, and let K_l be the true number of inversions between between the intersection skeletons at r and q along line l. Using the same notation as before, the approximation has the property that

$$K_l \in \left(K'_l - \frac{2K'_l}{\lfloor \tau^\varepsilon \rfloor}, K'_l + \frac{2K'_l}{\lfloor \tau^\varepsilon \rfloor}\right).$$

If K is the total number of inversions between r and q summed over all the lines in the intersection skeleton, then

$$K \in \left(K' - \frac{2K'}{\lfloor \tau^\varepsilon \rfloor}, K' + \frac{2K'}{\lfloor \tau^\varepsilon \rfloor}\right).$$

Choosing τ as before proves:

Theorem 6: Given a set of n hyperplanes H, the vertex in $A_d(H)$ with k^{th} largest (or smallest) x_1-coordinate can be found in $O(d\, n^{d-1} \log^{3/2} n)$ time.

FINAL REMARKS

There are basically two methods for designing geometric selection algorithms. The first technique was described by Chazelle [6], and the second is parametric search. Parametric search is effective as long as there is a sublinear-time ranking algorithm and some transformation from the selection problem to merging or sorting. Approximations may speed up this approach.

There are several interesting open problems in geometric selection. The most obvious one is to design an $O(n \log^{O(1)} n)$ time algorithm for L_2-interdistance selection. It should be noted that the general selection algorithm for L_2-interdistances is a generalization of the technique of Yao [20] for finding the largest L_2-interdistance. Another challenging open problem is to reduce the upper bound for selecting the largest L_2-interdistance. It seems likely that such a technique could be used for general selection as well. A third open problem is to use an approximation idea to speed up L_∞-interdistance selection; the difficulty here is that L_∞-interdistance ranking is far more complicated than counting inversions.

REFERENCES

1. A. Aggarwal, The 1986/87 Computational Geometry Days at Courant - Problems and Solutions, NYU, 1987.
2. M. Ajtai, J. Komlos and E. Szemeredi, An $O(n \log n)$ Sorting Network, *Proc. of the 15th Annual ACM Symp. on Theory of Computing*, 1983, pp. 1-9.
3. J. L. Bentley and M. I. Shamos, Divide-and-Conquer in Multidimensional Space, *Proc. of the 8th Annual ACM Symp. on Theory of Computing*, 1976, pp. 220-230.
4. M. Blum, R. W. Floyd, V. R. Pratt, R. L. Rivest and R. E. Tarjan, Time Bounds for Selection, *J. Computer and System Sciences*, **7**(4), 1973, pp. 448-461.
5. A. Borodin and J. E. Hopcroft, Routing, Merging and Sorting on Parallel Models of Computation, *Proc. of the 14th Annual ACM Symp. on Theory of Computing*, 1982, pp. 338-344.
6. B. Chazelle, New Techniques for Computing Order Statistics in Euclidean Space, *Proc. of the 1st Annual Symposium on Computational Geometry*, 1985, pp. 125-134.
7. R. Cole, Slowing Down Sorting Networks to Obtain Faster Sorting Algorithms, *J. ACM*, **34**(1), 1987, pp. 200-208.
8. R. Cole, J. S. Salowe, W. L. Steiger and E. Szemeredi, Optimal Slope Selection, *Proc. of the 15th International Colloquium on Automata, Languages and Programming*, 1988, pp. 133-146.
9. H. Edelsbrunner, J. O'Rourke and R. Seidel, Constructing Arrangements of Lines and Hyperplanes with Applications, *Siam J. on Computing*, **15**, 1986, pp. 341-363.
10. G. N. Frederickson and D. B. Johnson, The Complexity of Selection and Ranking in $X + Y$ and Matrices with Sorted Columns, *J. Computer and System Sciences*, **24**, 1982, pp. 197-208.
11. Z. Galil and N. Megiddo, A Fast Selection Algorithm and the Problem of Optimum Distribution of Effort, *J. ACM*, **26**, 1979, pp. 58-64.
12. D. E. Knuth, *The Art of Computer Programming. Volume III: Sorting and Searching*, Addison Wesley, Reading, MA, 1973.
13. N. Megiddo, Combinatorial Optimization with Rational Objective Functions, *Math. Oper. Res.*, **4**(4), 1979, pp. 414-424.
14. N. Megiddo, Applying Parallel Computation Algorithms in the Design of Serial Algorithms, *J. ACM*, **30**(4), 1983, pp. 852-865.
15. F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction*, Springer Verlag, New York, NY, 1985.
16. J. S. Salowe, *Selection Problems in Computational Geometry*, PhD Thesis, Rutgers University, 1987.
17. M. I. Shamos, Geometry and Statistics: Problems at the Interface, in *Algorithms and Complexity: New Directions and Recent Results*, J. F. Traub (ed.), Academic Press, New York, NY, 1976, 251-280.
18. L. G. Valiant, Parallelism in Comparison Problems, *Siam J. on Computing*, **4**(3), 1975, pp. 348-355.
19. D. E. Willard, New data structures for orthogonal range queries, *Siam J. on Computing*, 1985, pp. 232-253.
20. A. C. Yao, On Constructing Minimum Spanning Trees in *k*-Dimensional Space and Related Problems, *Siam J. on Computing*, **11**(4), 1982, pp. 721-736.

AN OPTIMAL ALGORITHM FOR CONSTRAINED DELAUNAY TRIANGULATION

DZ-MOU JUNG*
Electrical, Computer, and Systems Engineering Dept.
Rensselaer Polytechnic Inst., Troy, NY 12180

ABSTRACT

An $O(n \log n)$ algorithm to construct the Constrained Delaunay triangulation (CDT) of a planar straight-line graph first detects the difference between the standard Delaunay triangulation and the CDT, then constructs the constrained Delaunay edges within the difference portion.

INTRODUCTION

Applications of CDT include surface interpolation, finite element method, motion planning in the presence of polygonal obstacles in the plane. The definitions of (standard) Delaunay triangulation and Constrained Delaunay triangulation can be found in [7] and [3], respectively (see Fig. 1). Lee and Schacter [5] achieved the $O(n \log n)$ lower bound to construct a standard Delaunay triangulation. De Floriani et al. [3] provided an $O(n^2)$ incremental algorithm to construct the CDT with prespecified, possibly nonconvex boundary. Lee [6] presents two algorithms, one of which constructs the CDT with arbitrary obstacle edges in $O(n^2)$ time, while the other deals with the CDT of a simple polygon with all vertices on the boundary in $O(n \log n)$ time. Chew [2] and Wang [8] both achieved the $O(n \log n)$ lower bound in constructing the CDT. Chew constructs the CDT directly on the PSLG, while Wang constructs first the constrained Voronoi diagram, and then converts it to the dual graph, the constrained Delaunay triangulation, in $O(n)$ time. The method proposed below and described in greater detail in [4] is exactly the dual approach of [8].

ALGORITHM

```
Algorithm CONSTRUCT_CDT($V$, $E$)
1. Construct the standard Delaunay triangulation DT($V$) in $O(n \log n)$ time, e.g., using [5].
2. for each vertex $v$ in vertex set $V$ do
        Find the neighboring obstacle edges $B(v)$, the subset of the obstacle edge set $E$
        which are first encountered by a ray starting from $v$ parallel to either $X$ or $Y$ axis. $B(v)$
        contains at most 4 edges.
   Using the plan sweep method of [1], this step can be done in $O(n \log n)$ time. Only
   edges $(v_i, v_j)$ of DT($V$) which are intersected by some member of $B(v_i)$ or $B(v_j)$ are not
   edges of CDT($V$,$E$), so:
3. Discard every $(v_i, v_j)$ in DT($V$) crossed by any member of $B(v_i)$ or $B(v_j)$. This can be
   done in linear time.
4. for each $v \in V$ do
        Find $L(v)$, the set of obstacle edges closest to $v$ along some Delaunay edge incident to
        $v$.
        for each $l \in L(v)$ do Insert $v$ into $l$.difference polygon, in which every standard
            Delaunay edge is crossed by $l$ (Fig. 2a). This is the only portion of the original
            triangulation which must be modified.
   end for
   A closest obstacle edge of $v$ is not necessarily a neighboring obstacle edge $\in B(v)$ (see
   Fig. 3) but can be found from $B(v)$; see [8]. Again, this step can be done in $O(n \log n)$
   time by the plan sweep method of [1].
5. (a) for each difference polygon do
            Construct the interior Delaunay triangulation, using [6].
   (b) Merge several difference polygons sharing common edges which are neither standard
            nor constrained Delaunay edges (Fig. 2b).
   $O(n)$ obstacle edges, where $n = |V|$, can create $O(n)$ difference polygons. The interior CDT
   of a difference polygon with $n_i$ edges can be done in $O(n_i \log n_i)$ time [6]. The total
```

---

* This research was partially supported under NSF grant IRI-8704718

number of edges in $O(n)$ difference polygons are $\sum_i n_i = O(n)$, hence the total time to construct all CDTs is $\sum_i n_i \log n_i \le n \log n$. Merging two difference polygons can be done in linear time [6], so step 5 takes $O(n \log n)$ time.

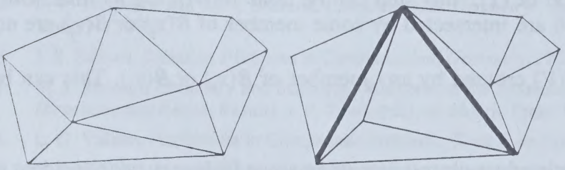
6. Replace the appropriate portion of the standard Delaunay triangulation by the retriangulated difference polygons.
   A single scan for each vertex can embed the difference edges in linear time.

CONCLUSION

We presented an optimal algorithm to construct the constrained Delaunay triangulation. Our algorithm is dual to [8] and does not compute the intermediate structure -- the standard and constrained Voronoi diagrams. Our method is suitable for the planar straight-line graph with relatively few obstacle edges. This is often the case in terrain models since only the dominant features, e.g., high and long ridges, rivers, and lake shores are desired to be preserved in the triangulation. If there are many obstacle edges in the graph, most of the standard Delaunay edges will be removed. An alternative approach starting from the obstacle edges and inserting the unknown Delaunay edges will possibly be more appropriate.

REFERENCES

1. J. Bentley and T. Ottmann, "Algorithms for Reporting and Counting Geometric Intersections," *IEEE Trans. on Computers*, Vol. c-**28** (1979), 643-647.
2. L. Chew, "Constrained Delaunay Triangulations," *Proc. 3rd annual symposium on computational geometry*, 1987, pp. 215-222.
3. L. De Floriani, B. Falcidieno, and C. Pienovi, "Delaunay-Based Representation of Surfaces Defined Over Arbitrarily Shaped Domains," *Computer, Vision, Graphics and image Proc.*, **32**, 1 (1985), 127-140.
4. D. Jung, "On Constructing the Constrained Delaunay Triangulation," Tech. Rep. 100, Electrical, Computer, and Systems Engineering Dept., Rensselaer Polytechnis Inst., 1988.
5. D. T. Lee and B. J. Schacter, "Two Algorithms for Constructing a Delaunay Triangulation," *Int. J. of Computer and Information Sciences*, **9**, 3 (1980), 219-242.
6. D. T. Lee and A. K. Lin, "Generalized Delaunay Triangulation for Planar Graphs," *Discrete Comput. Geom.*, **1** (1986), 201-217.
7. F. P. Preparata, M. I. Shamos, "Computational Geometry," Spring-Verlag, 1985.
8. C. Wang and L. Schubert, "An Optimal Algorithm for constructing the Delaunay triangulation of a set of line segments," *Proc. 3rd annual symposium on computational geometry*, 1987, pp. 223-232.



(a) (b)

Fig. 1 (a) A standard Delaunay triangulation;
(b) a Constrained Delaunay triangulation with heavy obstacle edges.

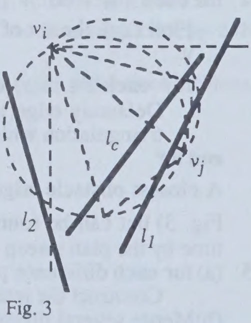
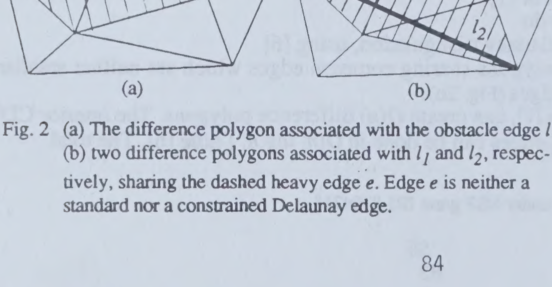


Fig. 3
l_c is a closest obstacle edge of v_i but $l_c \notin B(v_i)$, while $l_1, l_2 \in B(v_i)$ but neither is the closest obstacle edge of v_i along (v_i, v_j).



(a) (b)

Fig. 2 (a) The difference polygon associated with the obstacle edge l;
(b) two difference polygons associated with l_1 and l_2, respectively, sharing the dashed heavy edge *e*. Edge *e* is neither a standard nor a constrained Delaunay edge.

LOWER BOUNDS FOR SOME CONSTRAINED VORONOI DIAGRAMS

CAO AN WANG
Department of Computer Science
Memorial University of Newfoundland
St. John's, Newfoundland
Canada A1C 5S7

Abstract: In this paper, we first define a new Voronoi diagram of a set of points (called sites) in the presence of obstacles in the plane, where the obstacles are a set of line segments. We call the diagram a *constrained Voronoi diagram* . We then prove that constructing the constrained Voronoi diagram of n sites and m obstacles takes $\Omega(n^4)$ time and space in the worst case, where $m = c^*n$ and c is a constant; when n sites lie inside a simple polygon with $m \geq 3$ sides (called an m-gon), the construction of the diagram takes $\Omega(n^2)$ time and space in the worst case. We also prove that constructing the constrained Voronoi diagram of a set of n sites in the presence of m obstacles in three dimensional space takes $\Omega(n^6)$ time and space in the worst case, where the obstacles are a set of r-gons and $r \geq 3$ is a constant. When n sites lie inside a simple polyhedron with m facets, the construction of the diagram takes $\Omega(n^3)$ time and space in the worst case. We finally show that constructing the constrained Voronoi diagram of a set of n sites and m obstacles in k-dimensional space for a fixed k and $k > 3$ takes $\Omega(n^{2k})$ time and space in the worst case, where the obstacles are $(k-1)$-dimensional polytopes. In the concluding remarks, we discuss two degenerate problems.

SUMMARY

Voronoi diagrams are one of the important geometric structures in computational geometry which attracts a lot of attention [Aur84, Chew85, Fort86, Kirk79, Lee79, Sh78, Yap85].

Given a set of sites S in the plane, the *standard* Voronoi diagram of the set consists of a set of s Voronoi cells, $\{V(s_i)\}$ for $s_i \in S$, such that for any point $x \in V(s_i)$, $d(x, s_i) \leq d(x, s_j)$ for all $s_j \in S$, where $d(x,y)$ denotes the Euclidean distance between x and y.

There exist many variations from standard Voronoi diagrams. One of them is to consider diagrams in the presence of obstacles in the plane [Aro87, Wang87]. In this paper, we first consider the Voronoi diagram of a set of sites restricted by a set of obstacle line segments in the plane. In this diagram, all points in a Voronoi cell must be 'visible' from the site associated with that cell. Note that points of some area of the plane may not be visible from any site due to the blockage of obstacles, and hence the area does not belong to any Voronoi cell by definition. In general, a constrained Voronoi diagram may not cover the whole plane.

This characteristic of constrained Voronoi diagrams complicates their structures. In particular, a Voronoi cell may contain several disjoint sub-Voronoi cells. We shall show in Section

3 that the constrained Voronoi diagram of n sites and m obstacle line segments in the plane may contain $O(n^4)$ disjoint sub-Voronoi cells. If n sites lies inside a m-gon, then the diagram may contain $O(n^2)$ disjoint sub-Voronoi cells. We then study the constrained Voronoi diagram of n sites and m r-gons in three dimensional space and show the diagram may contain $O(n^6)$ disjoint sub-Voronoi cells. If n sites lie inside a polyhedron with m facets, the diagram may contain $O(n^3)$ disjoint sub-Voronoi cells. We finally show that the constrained Voronoi diagram of n sites and m $(k-1)$-dimensional polytopes in k-dimensional space for a fixed k may contain $O(n^{2k})$ disjoint sub-Voronoi cells. Therefore, we obtain the (output) lower bounds for consructing constrained Voronoi diagrams of the corresponding cases. In the concluding remarks, we further discuss some degenerate problems.

Acknowledgement: Thanks R. Seidel for proposing and discussing the problem.

References

[Aro87] Aronov B., (1987) "On the geodesic Voronoi diagram of point sites in a simple polygon", *Proc. of 3rd ACM Symposium on Computational Geometry* , Waterloo, Canada,pp.39-49.

[Aur84] Aurenhammer F. and Edelsbrunner H., (1984), "An optimal algorithm for constructing the weighted Voronoi diagram in the plane", *Pattern Recognition* , 17(2), pp.251-257.

[Avi83] Avis D. and Bhattacharya B., (1983), "Algorithms for computing d-dimensional Voronoi diagrams and their duals", *Advances in Computing Research* , Edited by Preparata F., 1,JAI press, pp.179-209.

[Bro79] Brown K., (1979), "Voronoi diagrams from convex hulls", *The Information Processing Letters* , Vol.9, No.5, pp.223-228.

[Chaz85] Chazelle B. and Edelsbrunner H.,(1985), "An improved algorithm for constructing k-th order Voronoi diagrams", *Proc. of 1st ACM Symposium on Computational Geometry* , pp.228-234.

[Chew85] Chew L., Drysdale R., (1985), "Voronoi diagrams based on convex distance functions", *Proc. of 1st ACM Symposium on Computational Geometry* , pp.235-244.

[Ede86] Edelsbrunner H.,O'Rourke J., Seidel R., (1986), "Constructing arrangements of lines and hyperplanes with applications", *SIAM J. Computing.* , Vol.15, No.2, pp.341-363.

[Fort86] Fortune S. (1986), " A sweepline algorithm for Voronoi diagrams", *Proceedings of thee 2nd Symposium on Computational Geometry* , NY, pp.313-322.

[Gui85] Guibas L., Stolfi J., (1985), "Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams", *ACM Trans. on Graphics* , Vol.4, No.2, pp.74-122.

[Kirk79] Kirkpatrick D., (1979), "Efficient computation of continuous skeletons", *Proceedings of the Twentieth Annual Symposium on the Foundations of Computer Science* , pp. 18-27.

[Lee84] Lee D. T., Preparata F., (1984), "Computational geometry - a survey", *IEEE Trans. On Computers* , Vol. C-33, pp. 1072-1101.

[Prep85] Preparata F., Shamos M.,(1985), " *Computational Geometry* ", springer Verlag.

[Sei82] Seidel R., (1982), "The complexity of Voronoi diagrams in higher dimensions", *Proc. 20th Allerton Conference on Comm., Control and Comput.* , pp.94-95.

[Tou82] Toussaint G., Avis D., (1982), "On a convex hull algorithm for polygons and its applications to triangulation problems", *Pattern Recognition* ", Vol. 15, pp 23-29.

[Wang 87] Wang C. and Schubert L,, (1987), "Optimal algorithm for constructing the Delaunay triangulation of a set of line segments", *Proc. of 3st ACM Symposium on Computational geometry* , pp. 223-233.

[Yap 85] Yap C. (1985), "An O(n log n) algorithm to construct the Voronoi diagram of a set of simple curves", Technical report, NYU.

A UNIFIED APPROACH TO FINDING ENCLOSING AND ENCLOSED TRIANGLES

DAVID M. MOUNT[1]
Department of Computer Science and
Institute for Advanced Computer Studies
University of Maryland

SHARAT CHANDRAN[2]
Center for Automation Research
University of Maryland

Abstract

The problems of finding the maximum area triangle enclosed in and the minimum area triangle enclosing a given n-sided convex polygon are both known to be solvable in optimal $O(n)$ time. The algorithms used to solve these problems are quite different in their approaches. In this paper we present a unified approach for solving both problems. In particular, we show that both the maximum enclosed triangle and the minimum enclosing triangle of a convex polygon can be determined from a set of $O(n)$ enclosing triangles, which we call P-stable triangles. We show how to compute these triangles in $O(n)$ sequential time, and in $O(\log n)$ time on a CREW PRAM with n processors. The parallel algorithm for maximum enclosed triangles is an improvement over the existing $O(\log^2 n)$ algorithm.

1 Introduction

The problems of finding minimum area enclosing and maximum area enclosed k-gons arise from the desire to approximate many sided convex polygons by polygons with fewer sides. One application is in the area of collision avoidance in robotics [4, 5]. It is easier to test interference of objects with fewer sides, and once interference is established, a more refined analysis can be made. Algorithms for finding maximal enclosed and minimal enclosing k-gons polygons were presented in [3]. Unfortunately these algorithms are not known to be optimal. The case of finding the minimum enclosing and maximum enclosed triangles ($k = 3$) are the only cases where optimal sequential algorithms are known. An $O(n)$ algorithm for computing the largest area triangle enclosed in a given convex polygon was given by Dobkin and Snyder [6]. Here n is the number of vertices in the polygon. The problem of finding the minimum area triangle enclosing a given convex polygon was first considered by Klee and Laskowski [7] where an $O(n \log^2 n)$ algorithm was given. This was subsequently improved to $O(n)$ by O'Rourke, Aggarwal, Maddila and Baldwin [9].

Both linear time sequential algorithms are roughly based on the method of "rotating calipers" [10] to generate a finite set of intermediate triangles from which the optimum is selected. In spite of the apparent similarity between the two problems, the details of the algorithms and their proofs of correctness are very much different. As an indication of the disimilarity, an $O(\log n)$ time and $O(n)$ processor parallel algorithm was given by Aggarwal, Chazelle, Guibas, O'Dunlaing and Yap [1], however they state that their methods do not yield a better than $O(\log^2 n)$ parallel algorithm for the problem of finding the maximum enclosing triangle. The model of parallel computation is a CREW PRAM, a parallel shared memory machine with concurrent-read and exclusive-write.

---

[1] Address: Dept. of Computer Science, University of Maryland, College Park, MD 20742.

[2] Address: Center for Automation Research, University of Maryland, College Park, MD 20742. The support of the FMC Corporation is gratefully acknowledged.

Our initial goal in this study was to find an efficient parallel algorithm for the problem of finding the maximum enclosed triangle. In the process, we discovered that these two problems are much more closely related than previously thought, in the sense that both can be solved easily once a particular set of $O(n)$ intermediate triangles, which we call *P-stable triangles*, have been computed. These intermediate triangles are a superset of the P-anchored triangles introduced by Klee and Laskowski for finding minimum enclosing triangles [7]. In this paper we prove that given a convex polygon P, the set of P-stable triangles can be computed in $O(n)$ sequential time and $O(\log n)$ parallel time on a CREW PRAM with n processors. We show that by generating these triangles we can determine the family of all minimum enclosing and maximum enclosed triangles for a convex polygon. As a consequence, we have improved the parallel complexity of finding maximum enclosing triangles

2 Preliminaries

Throughout this paper, when referring to maximum and minimum triangles we mean triangles of maximum or minimum area. Throughout, P will denote a convex polygon with n vertices. When referring to P we mean the boundary of P formed from the clockwise circular sequence of its vertices and edges. For two distinct points a and b on P, we use the notation (a, b) to denote the open interval of points lying strictly between a and b on P clockwise from a to b, and $[a, b]$ is the closure of this interval. Note that a and b need not be vertices, that is, they may lie on the interior of edges of P

It is easy to show that the vertices of any maximum triangle enclosed in a convex polygon must contact the polygon's boundary, as must the edges of any minimum enclosing triangle. We will assume that any triangle that encloses P or is enclosed in P is represented so that its points of incidence with P (vertices or edges) can be determined in $O(1)$ time. We say that a triangle that encloses P is *flush* with P if some edge of P is a subsegment of one of the sides of the triangle. A line L is said to *support* a convex polygon P if P and L intersect, and P lies entirely in one of the closed halfspaces defined by L. We begin with two lemmas that summarize the local properties of maximum enclosed and minimum enclosing triangles.

LEMMA 2.1 (Dobkin and Snyder)

(i) If T is a maximum enclosed triangle for a convex polygon P then for each vertex v of T, the line drawn through v and parallel to the opposite side of T is a line of support for P.

(ii) Given a convex polygon P, there exists a maximum enclosed triangle in P whose vertices are a subset of the vertices of P.

LEMMA 2.2 (Klee and Laskowski)

(i) If T is a minimum enclosing triangle for P then the midpoint of each side of T intersects P.

(ii) If T is a minimum enclosing triangle then at least one side of T is flush with P.

We now introduce the notions of P-supported and P-stable triangles.

DEFINITION: Given a convex polygon P, an enclosing triangle is *P-supported* if the midpoints of two of its sides, called the *legs*, intersect P. (The third side, called the *base*, intersects P arbitrarily.)

When referring to a P-supported triangle, ABC, we will adopt the naming convention that C is the vertex opposite the base. We will assume throughout that P-supported triangles are represented so that we know which sides are its legs and which side is its base. It follows from elementary geometry that the midpoints a and b of the legs AC and BC, respectively, form a line segment that is parallel to and of one half the length of the base AB.

Given the line L supporting P, there is, in general, an infinite family of P-supported triangles whose base lies on the line L. There is a sense in which all such triangles are equivalent, which was observed earlier [9].

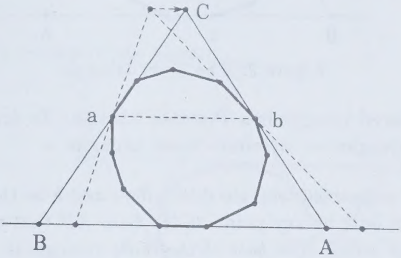


Figure 1: The canonical P-supported triangle.

LEMMA 2.3 *Given a convex polygon P and a supporting line L, let S denote the set of P-supported triangles whose bases lie on L.*

(i) S is nonempty, and all the elements in S have the same area, and their corresponding legs share the same points as midpoints.

(ii) If S contains more than one element then the midpoints a and b are vertices of P, and the set of apexes of S form a line segment, parallel to L, and at twice the height above L as the line segment ab. The endpoints of this segment can be found in $O(1)$ time (see Figure 1).

Since we are interested in the areas of enclosing triangles, any local transformation that preserves the area of a P-supported triangle will not affect the final output. Thus, given a P-supported triangle, in $O(1)$ time we can select one endpoint (the rightmost, say) of the set of allowable apexes to form the *canonical* P-supported triangle having this base direction (see Figure 1).

The computational difficulty of P-supported triangles (even in canonical form) is that there are an infinite number of them, corresponding to each possible line of support for P. To limit the set to a finite collection of triangles, but encompassing enough to contain all locally minimum enclosing triangles, Klee and Laskowski introduced the notion of a P-anchored triangle [7, 9]. Such a triangle is simply a P-supported triangle whose base is flush with P. One notable feature of the definition of a P-anchored triangles is its asymmetry, in that the anchoring condition applies only to the base edge and not to the legs. In the case of minimum enclosing triangles this condition is sufficient to solve that problem. There are a few ways to generalize the anchoring condition so it can applied to both enclosing and enclosed triangles. The most useful one is to require that the midpoint of one of the legs is a vertex of P. This gives us the notion of a P-stable triangle.

DEFINITION: A P-supported triangle is *P-stable* if either

(i) the base is flush with P, or

(ii) one of the legs has as its midpoint a vertex of P.

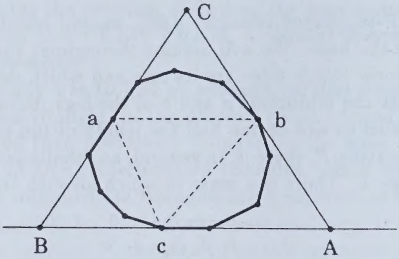


Figure 2: The inner triangle.

Clearly every P-anchored triangle is a P-stable triangle. To draw the connection between enclosing and enclosed triangles we introduce inner triangles.

DEFINITION: Given a P-supported triangle ABC, let a and b be the midpoints of the legs BC and AC, respectively, and let c be any point on the base AB that also lies on P. The triangle abc is an *inner triangle* of ABC. The *base* of the inner triangle is the edge ab, the remaining sides are the *legs*, and c is the *apex* (see Figure 2).

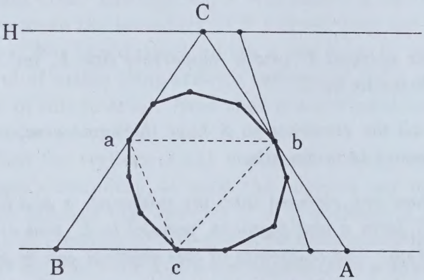


Figure 3: Generating an outer triangle from an inner triangle.

If the base of a P-supported triangle is flush with an edge of P, then there are and infinite number of inner triangles all of equal area, and hence we will adopt the convention of selecting the furthest eligible clockwise point for the apex of the inner triangle. Thus we can talk about "the" inner triangle for a given P-supported triangle. A number of simple observations regarding inner triangles can be made at this point. The first observation follows from Lemma 2.3, and the other two follow from elementary geometry.

LEMMA 2.4 *Let P be a convex polygon, let T be a P-supported triangle, and let t be the inner triangle for T. Then*

(i) t is uniquely determined from the angle of T's base,

(ii) the bases of T and t are parallel, and

(iii) the area of T is four times the area of t.

It is obvious how to generate the unique inner triangle for a given P-supported triangle (given P). If it is known that abc is an inner P-supported triangle, the unique *canonical* outer triangle ABC can be found in $O(1)$ time as follows. Think of ab as being horizontal directed left to right. The base AB is parallel to ab and passes below through c. The apex C lies on a line H at twice the height of ab above this base. Extend the edges of P lying just clockwise from a and b until they intersect H. The leftmost of these two intersection points is the apex C (see Figure 3). (If the line from C through b is not a line of support for P then it follows that abc cannot be an inner triangle.)

The fundamental observation linking the two problems of finding minimum enclosing and maximum enclosed triangles is the following.

LEMMA 2.5

(i) There exists a minimum enclosing triangle that is a P-stable triangle.

(ii) There exists a maximum enclosed triangle that is an inner triangle of some P-stable triangle.

PROOF: Omitted from this version of the paper. □

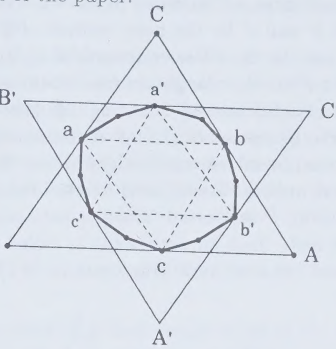


Figure 4: Interspersing property.

Thus, to determine the minimum enclosing triangle it suffices to compute the set of canonical P-stable triangles (which we show below to be a finite set) and to select the triangle of minimum area. To determine the maximum enclosed triangle it suffices to select the P-stable triangle of maximum area, and then compute its corresponding inner triangle (since their areas are linearly related). The two principal tasks remaining are to prove that there are not "too many" P-stable triangles, and to show how to compute them all. Both tasks rely on the following *interspersing property*, that seems to be pervasive throughout the theory of enclosing and enclosed polygons [6, 9, 3]. This property states that as the orientation of the base edge advances in a clockwise direction, the points of the corresponding inner triangle move clockwise about P. This lemma is not stated in exactly this form in [9], but it follows directly from the proof of their Lemma 2, together with the simple observation that as the orientation of the base turns clockwise, the point of contact with the base, c, also moves clockwise around P.

LEMMA 2.6 (O'Rourke, Aggarwal, Maddila, and Baldwin)
Let ABC be a P-supported triangle with base AB and inner triangle abc. Let $A'B'C'$ be a P-supported triangle whose base $A'B'$ lies between AB and BC in angular orientation, and let $a'b'c'$ be its inner triangle. Then $c' \in [c, a]$, $b' \in [b, c']$ and $a' \in [a, b']$. (See Figure 4.)

From the interspersing lemma we immediately derive the following observation, which establishes that the number of P stable triangles is linear in the size of P.

LEMMA 2.7 *Given a convex polygon P with n vertices, there are at most $3n$ canonical P-stable triangles.*

PROOF: Omitted from this version of the paper. □

3 Sequential Algorithm

It follows from Lemma 2.5 that in order to find the maximum enclosed and minimum enclosing triangles it suffices to generate all $O(n)$ canonical P-stable triangles. As in [9], our algorithm is an example of the rotating calipers technique, generating the all P-anchored triangles in clockwise order according to the orientation of the base edge. Due to the more general class of triangles being generated, the algorithm is somewhat more involved than the algorithm appearing in [9].

The algorithm begins by generating one P-anchored triangle ABC starting with any base edge of P. This can be done in $O(log^2 n)$ time by the algorithm given in [7]. Transform ABC into canonical form if it is not already in this form. Compute the corresponding inner triangle abc in $O(1)$ time. Recall that c is a vertex of P, but a and b need not be vertices. Let e_a, e_b, and e_c denote the edges of P lying on and immediately clockwise of points a, b and c, respectively. Since ABC is in canonical form, at least one of the sides of BC or CA is flush with e_a or e_b respectively. Let a' and b' and c' be the next vertices of P in clockwise order after a, b and c. That is, a', b' and c' are the clockwise endpoints of e_a, e_b and e_c, respectively. We describe how to generate the next P-stable triangle ordered clockwise by the angle of the base in $O(1)$ time. It follows from Lemma 2.7 that by iterating this process at most $3n$ times, we will have returned to the initial triangle, generating all P-stable triangles.

We consider three cases, based on which edges of the current triangle ABC are flush with edges of P. The argument in each case is based on the idea of gradually rotating the base edge and considering the resulting P-supported triangles until one of the anchoring conditions for a P-stable triangle is first met. Such an occurrence is called an *anchoring event*. In each case, we show how to determine the next anchoring event in $O(1)$ time.

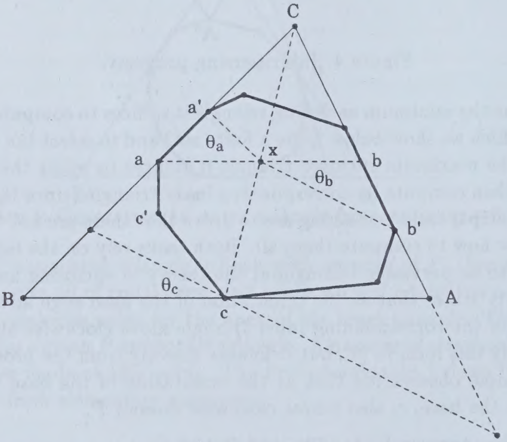


Figure 5: The case of two flush legs.

(Case 1: BC is flush with e_a and CA is flush with e_b.) Let x denote the midpoint of the segment Cc. Since a and b are midpoints of BC and CA, it follows that x is also on the segment ab. Intuitively, as the angle of the base rotates clockwise through a small angle, the resulting P-supported triangle will have a base touching the vertex c, whose midpoints lie within the (nonempty) intervals $[a, a']$ and $[b, b']$, implying that the legs remain flush with e_a and e_b. Hence, the apex does not move. Since x is the midpoint of Cc, the line connecting these two midpoints must pass through x and will be at the same angle as the base edge. The next anchoring event occurs either when the base rotates so far as to become flush with e_c, or when the line between the triangle midpoints passes through one of the vertices a' or b'. With this in mind, we compute the clockwise angles corresponding to these events: $\theta_c = \angle Bcc'$, $\theta_a = \angle axa'$ and $\theta_b = \angle bxb'$ (see Figure 5).

Let $\theta = \min(\theta_a, \theta_b, \theta_c)$. The next P-stable triangle is found by rotating the base clockwise through the angle θ. As mentioned above, the base of the resulting triangle passes through c with opposite vertex C, and the sides are flush with e_a and e_b. If $\theta = \theta_a$ or $\theta = \theta_b$ then the midpoints of this triangle touch the corresponding vertices, and so e_a or e_b must be updated. In either of these cases the triangle may not be in canonical form, and if not convert it into canonical form in $O(1)$ time. If $\theta = \theta_c$, the triangle is flush with e_c, and so c and e_c must be updated.

(Case 2: BC is flush with e_a, but CA is not flush with e_b.) Due to space limitations, we omit this case from this version of the paper. The method similar to Case 1, but takes the point b as being fixed.

(Case 3: CA is flush with e_b, but BC is not flush with e_a.) This case is symmetric to Case 2 with the roles of A, a and a' swapped with B, b, and b' respectively, with the only exception that θ_c is still defined to be the angle $\angle Bcc'$.

The above algorithm runs in $O(1)$ time per case. The total time needed to generate all the P-stable triangles is $O(n)$. As mentioned above, once the P-stable triangles have been generated, the largest and smallest can be found for determining the largest enclosed triangle and the smallest enclosing triangle, respectively. As a matter of practicality, the algorithm would actually just maintain the first triangle, the current triangle, and simply keep track of the largest and smallest triangles encountered thus far. Thus the algorithm needs only $O(1)$ working storage.

4 Parallel Algorithm

In this section we present a parallel algorithm that computes all of the $O(n)$ P-stable triangles. The model of parallel computation that we employ is a CREW PRAM, a concurrent-read, exclusive write parallel random access machine. Given n processors, our algorithm achieves a running time of $O(\log n)$. The algorithm is similar in structure to the $\sqrt{n}$-divide and conquer algorithm presented in [1] for solving the restricted problem of finding all the P-anchored triangles, but the "back-and-forth subdivision" used by their algorithm has been replaced by simpler redundancy scheme (owing in part to the more symmetric nature of P-stable triangles over P-anchored triangles). Once all of the P-stable triangles have been computed, it is an easy matter to determine the triangles of maximum and minimum area in $O(\log n)$ time.

The algorithm operates in a series of $O(\log \log n)$ stages. Intuitively, the algorithm exploits the interspersing lemma, Lemma 2.6, at each stage to compute the P-stable triangles. During successive stages we compute ever larger subsets of the P-stable triangles whose corresponding inner triangles are interspersed according to Lemma 2.6. Thus later stages can refine their search in the ever narrowing gaps between cyclically adjacent inner triangles. This provides us with the sort of balance of effort seen in parallel algorithms, where later stages produce more output, but each search is computationally easier by a proportional amount.

The key to the efficiency of the algorithm will be the need to guarantee that between any cyclically adjacent pair of triangles at each stage, there are at most a small number (made precise below) of vertices of P. To see the difficulty in maintaining this condition, consider the case of two inner triangles, abc and $a'b'c'$. Although, by judiciously choosing a' to be close to a, we can guarantee that the number of vertices of P in the range $[a, a']$ is small, we cannot infer that the number of vertices of P within $[b, b']$ and $[c, c']$ are correspondingly small. In order to circumvent this problem, we use a redundancy scheme, by producing three new triangles, one whose first vertex is close to a, one whose second vertex is close to b, and one whose third vertex is close to c. Irrespective of how these three triangles intersperse, this will guarantee that there are a small number of vertices between every pair of adjacent inner triangles.

To make this idea more formal, for each stage s, $s \geq 1$, let $n_s = \left\lfloor n^{1/2^s} \right\rfloor$. Thus, $n_1 = \lfloor \sqrt{n} \rfloor$, $n_2 = \lfloor \sqrt[4]{n} \rfloor$, and so on. Define V_s to be the subset of vertices consisting of every n_s-th vertex in P. We will maintain the invariant that after stage s, there will be $O(n_s)$ vertices of P lying cyclically between any two adjacent interspersed P-stable triangles that have been generated so far. After $O(\log \log n)$ stages, we will have computed a P-stable triangle anchored (in every possible way) at every vertex, implying that we have computed all the P-stable triangles.

To define how these inner triangles are generated we will need to introduce the notion of a P-stable triangle that is anchored in a particular way to a side of P and investigate how to generate these triangles. We say that a P-stable triangle is *a-anchored* at a vertex v, if it is flush with the edge e whose clockwise endpoint is v, and if its inner triangle abc satisfies $a = v$. (Recall our naming convention for inner triangles.) Note that an a-anchored triangle will not generally be in canonical form. The notions b-anchored and c-anchored are defined similarly.

In previous works, it was discussed in detail how to find c-anchored triangles [7, 9]. In particular, the following result was given in [7] in a restricted form for edges rather than points.

LEMMA 4.1 *Given a convex polygon P and a vertex v on P, there exists a unique (canonical) P-stable triangle T that is c-anchored at v. Furthermore, given v, each of the endpoints of the corresponding inner triangle can be found in $O(\log^2 n)$ time by a binary search that uses $O(\log n)$ time per probe.*

PROOF: Omitted from this version of the paper. □

This Lemma provides the basis for Klee and Laskowski's $O(n \log^2 n)$ algorithm for finding the smallest enclosing triangle, since from Lemma 2.2, it suffices to determine the P-supported triangle that is c-anchored at each vertex. We also need supplementary results on finding a-anchored and b-anchored triangles. We will state the result for a-anchored triangles, and an analogous result holds for of b-anchored triangles by a symmetric argument.

LEMMA 4.2 *Given a convex polygon P and a vertex v on P, there exists a unique (canonical) P-stable triangle that is a-anchored at v. Furthermore, given v, each of the endpoints of the corresponding inner triangle can be found in $O(\log^2 n)$ time by a binary search that uses $O(\log n)$ time per probe.*

PROOF: Omitted from this version of the paper. □

We now have the necessary machinery to describe the parallel algorithm. Let the current stage be $s \geq 1$. We maintain the invariant that at the end of stage s, for each vertex $v \in V_s$ we compute an a-anchored, b-anchored and c-anchored triangle at v. Thus there will be $3|V_s| = 3n/n_s$ triangles computed altogether by the end of stage s (not all necessarily distinct). These triangles are maintained in cyclic order according to the interspersing lemma. For each vertex in $v \in V_s$ we allocate n_s processors with which to compute a-, b-, and c-anchored triangles at v. This implies that n processors are used altogether at any stage. As mentioned earlier, within $O(\log \log n)$ stages, we will have computed all the P-stable triangles. The largest and smallest of these triangles can be determined easily in $O(\log n)$ time using n processors.

At the beginning of stage s the cyclically ordered inner triangles generated so far define three separate partitions of the boundary of P into closed intervals. In particular, if $a_0, a_1, \ldots, a_{m-1}$ are the a-vertices of the inner triangles generated so far, the a-partition consists of the half-closed intervals $[a_i, a_{i+1})$ (indices taken modulo m). (Initially, all three partitions are trivial, consisting of a single interval.) There is an analogously defined b-partition and a c-partition. It follows from the interspersing lemma, that the inner triangle of any subsequent P-stable triangles will lie within the closures of some three intervals corresponding to cyclically adjacent triangles, $[a_i, a_{i+1}]$, $[b_i, b_{i+1}]$, and $[c_i, c_{i+1}]$. Each a-interval (an interval of the a-partition) has at most n_{s-1} vertices of P, since we explicitly constructed a-anchored triangles at the points of V_{s-1} during the previous stage. Similar bounds bold for the b- and c-intervals. Thus, by the interspersing lemma, as more and more triangles are generated, the location of subsequent inner triangles becomes more and more constrained.

The data are stored as follows. Each cyclically adjacent pair of inner triangles abc and $a'b'c'$ that have been generated so far define three corresponding intervals $[a, a')$, $[b, b')$, and $[c, c')$, one in each of the three partitions. These three intervals are linked to each other, and each vertex of P contains three pointers to the intervals in each of the three partitions containing this vertex. Vertices of inner triangles are labeled as either a-, b-, or c-vertices.

The processing at stage s is as follows. Each vertex $v \in V_s - V_{s-1}$ is allocated n_s processors with which to compute its a-, b-, and c- anchored triangles. Let us consider the case of finding the a-anchored triangle, since the other two cases are similar. We determine the a-interval $[a, a')$ that contains v and access the corresponding b- and c-intervals $[b, b')$ and $[c, c')$. Applying Lemma 4.2, we use a parallel binary search through the closures of the b- and c-intervals to locate the a-anchored triangle and compute its corresponding inner triangle.

Once the triangles have been computed, we need to update the ordered set of triangles, and hence refine the three partitions. First note that the number of vertices of V_s in any interval is at most $n_{s-1}/n_s = n_s$. Let us consider the case of an a-interval, since the other cases are similar. By the interspersing lemma, there are three new sets of inner triangle endpoints that might fall into the interval $[a, a')$. They are the a-vertices of each of the following: (1) the a-anchored triangles generated in $[a, a')$, (2) the b-anchored triangles in $[b, b')$, and (3) the c-anchored triangles in $[c, c')$. We can maintain three sorted lists containing these a-vertices. Each list is associated with the sorted list of vertices of V_s lying within each of the three intervals that was responsible for generating this triangle vertex. Each list can have at most n_s elements in it. Using the processors allocated to one of the vertices of V_s in one of the three intervals, we merge these three lists into one sorted list and replace the single interval $[a, a')$ with the refined list of intervals. (Note that at least one of the three intervals must contain a vertex of V_s, for otherwise no new triangles would be generated in any of the intervals.) Each vertex of P in $[a, a')$ is given a new label indicating its new interval membership. This is done by parallel prefix of the vertices of P in $[a, a')$, using the set of processors allocated to all of the three intervals. Assuming that there is at least one vertex of V_s in either of the three intervals, the number of processors is proportional to the number of vertices of P in $[a, a')$.

To analyze the running time of stage s, we make use of the following facts that are well known from the theory of parallel algorithms.

LEMMA 4.3 *(i) Consider a sorted list a_i, $1 \le i \le n$, and a value x, $a_1 \le x < a_n$. Let $Q(n)$ denote the sequential time needed to access an arbitrary member a_i. We can find the index i such that $a_i \le x < a_{i+1}$ in $O(Q(n))$ time on a CREW PRAM with $\sqrt{n}$ processors [8].*

(ii) Two sorted lists of size n can be merged in $O(\log\log n)$ time on a CREW PRAM using n processors [2].

(iii) The parallel prefix of a list of size n can be computed in $O(\log n)$ time on a CREW PRAM using n processors.

In order to find the anchored triangles, we perform a parallel binary search through a list of size n_{s-1} using $n_s = \sqrt{n_{s-1}}$ processors. From Lemmas 4.1 and 4.2, it takes $O(\log n_{s-1})$ time for each access, and hence by Lemma 4.3(i) the time required to locate the triangle is $O(\log n_{s-1})$. To merge the lists of triangle endpoints as part of the interval refinement process, we need to merge two lists of size at most $O(n_s)$ using $O(n_s)$ processors. By Lemma 4.3(ii) $O(\log\log n_s)$ time suffices. Updating the labels on each vertex indicating its interval membership can be performed in $O(\log n_{s-1})$ time by Lemma 4.3(iii). Hence the total time for stage s is

$O(\log n_{s-1})$. The time for the entire algorithm is bounded by

$$\begin{aligned}\sum_{s\geq 1} \log n_{s-1} &= \sum_{s\geq 0} \log n_s \\ &= \sum_{s\geq 0} \log n^{1/2^s} \\ &= \sum_{s\geq 1} \frac{\log n}{2^s} = O(\log n).\end{aligned}$$

References

1. A. Aggarwal, B. Chazelle, L. Guibas, C. Ó'Dúnlaing, C. Yap, "Parallel computational geometry," *Proc. of the 25th Annual Symp. on Foundations of Computer Science*, 1984, 468–477.

2. A. Borodin and J. Hopcorft, "Routing and merging and sorting on parallel models of computation," *J. Comput. System Sci.*, 30 (1985), 130–145.

3. J.E Boyce, D. P. Dobkin, R. L. Drysdale and L. J. Guibas, "Finding extremal polygons," *SIAM J. Comput.*, 14 (1985), 134–147.

4. J. S. Chang and C. K. Yap, "A polynomial time solution for potato-peeling and other polygon inclusion and enclosure problems," *Proc. of the 25th Annual Symp. on Foundations of Computer Science*, 1984, 408–416.

5. N. A. A. DePano and A. Aggarwal, "Finding restricted k-envelopes for convex polygons," *Proc. of the 22nd Allerton Conf. on Comm., Contr. and Comp.*, 1984, 81–90.

6. D. P. Dobkin and L. Snyder, "On a general method for maximizing and minimizing among certain geometric problems," *Proc. of the 20th Annual Symp. on Foundations of Computer Science*, 1979, 9–17,

7. V. Klee and M. C. Laskowski, "Finding the smallest triangles containing a given convex polygon," *Journal of Algorithms* 6 (1985), 359–375.

8. C. Kruskal, "Searching, merging and sorting in parallel computation," *IEEE Trans. Comput.*, C-32 (1983), 942–946.

9. J. O'Rourke, A. Aggarwal, S. Maddila and M. Baldwin, "An optimal algorithm for finding maximal enclosing triangles," *Journal of Algorithms* 7 (1986), 258–269.

10. G. T. Toussaint, "Solving geometric problems with the 'rotating calipers,' " *Proc. of IEEE MELECON 83* (1983).

COVERING ORTHOGONAL POLYGONS WITH SQUARES

LARRY J. AUPPERLE
Dept. of Computer Science, Princeton U., Princeton, NJ 08544
HAROLD E. CONN
Dept. of Computer Science, Johns Hopkins U., Baltimore, MD 21218
J. MARK KEIL
Dept. of Computational Sci., U. of Saskatchewan, Saskatoon, Canada S7N 0W0
JOSEPH O'ROURKE
Dept. of Computer Science, Smith College, Northampton, MA 01063

ABSTRACT

The problem of covering an orthogonal polygon with a minimum number of squares is shown to be solvable in polynomial time for simply-connected polygons, making this one of the few minimum covering problems known to be tractable. The polynomial algorithm is based on establishing that a particular graph associated with the problem is chordal, a subclass of perfect graphs for which efficient clique covering algorithms are known. For polygons with holes, the minimum covering problem is shown to be NP-complete. The square cover problem finds application in image processing, where maximal squares are used to compress pictures into efficient data structures.

1. INTRODUCTION

We consider the problem of finding a minimum square cover of an orthogonal polygon. An *orthogonal polygon* is a polygon having all its edges parallel to either of two given orthogonal directions. It may contain holes that are themselves orthogonal polygons. We restrict our consideration to orthogonal polygons having all vertices at integral coordinates. These orthogonal polygons are also referred to as *boards*. A unit square with vertices at integral coordinates is called a *block* *. A *square* is a square subset of the blocks of a given board. A *square cover* of a board is a collection of squares each of which is a subset of the board and such that the set union of the collection of squares is equal to the board.

In this paper we analyze the complexity of the minimum square cover problem first for simply connected boards and then for boards containing holes. Given a board B, we associate with it a graph $G(B)$ as follows. Each block in the board B corresponds to a node in the graph $G(B)$. There is an edge between two nodes in $G(B)$ iff the corresponding blocks in B belong to a square of B. We show that the graphs associated with simply-connected boards are chordal graphs. This is a strengthening of a result due to Albertson and O'Keefe. In [2] they showed that these graphs are perfect graphs. Chordal graphs form a subclass of perfect graphs, and using a known algorithm for finding a minimum clique cover of a chordal graph [4], we can find a minimum square cover of a board with no holes in polynomial time, $O(N^{2.5})$ for boards with N blocks.

In [1] Aupperle improves the complexity of the algorithm to $O(rc \ min(r, c))$, where r and c are the number of rows and columns of blocks in the board. This complexity is $O(N^{1.5})$ in the worst case.

When a board is allowed to contain holes, we show that the problem of finding a minimum square cover is NP-complete.

This problem is interesting for two reasons, one theoretical and one practical. On the theoretical side, there has been considerable interest in geometric minimum covering problems, but unfortunately most of the most natural problems are NP-complete [7]. Just recently it was shown by Culberson and Reckhow [3] that one of the more interesting unsettled cases, minimum covers of simple orthogonal polygons by rectangles, is also NP-complete. Thus minimum coverage by squares turns out to be one of the few of these covering problems that can be solved in polynomial time.

Coverage by squares also has practical applications. The *medial axis* (also called the *symmetric axis* or *skeleton*) of a polygon is the locus of centers of maximal disks contained in the polygon. When specialized to the L_∞ metric for applications to digital images, the medial axis is the locus of centers of maximal squares of odd side length [8]. The digital medial axis transform (*MAT*) is used for picture compression: simple images may be covered by few squares, and easily reconstructed from the *MAT* [10]. The image processing literature has assumed that the NP-completeness results for coverage by rectangles obviated any fast algorithm for coverage by squares [10]; thus no attempt was made to find *minimum* covers. Our results show that in the important special case of a connected image region, pessimism was unwarranted: a minimum cover can be found quickly.

---

* For image processing applications, a board is a picture or an image, and a block is a pixel.

Additionally, Scott and Iyengar [9] have defined the *TID*, Translation Invariant Data Structure, a method for representing images. The *TID* for a given image consists of a list of maximal squares covering all pixels within the image, sorted by certain position and size criteria. In order to reduce the cost of storing and manipulating a *TID*, the underlying list of maximal squares should be as small as possible.

2. COVERING A SIMPLY-CONNECTED BOARD

In this section we prove that if B is a simply connected board, then $G(B)$ is a chordal graph. A graph is said to be *chordal* if it has the property that every (simple) cycle of length strictly greater than 3 possesses a chord. Before proceeding with the proof, some definitions and a lemma from [2] are needed. Given a board B and a square S in that board, S is said to be *maximal* if it is not contained in any larger square of the board B. A maximal square S is said to be *disconnecting* if the interior of $B-S$ is not connected. A block of B is a *boundary* block if it intersects the boundary of B in at least one edge. A block in a square S is a *border* block of S if either: (a) it shares an edge with a block of $B-S$, or (b) it is a boundary block. A $1 \times p$ rectangular subset of the blocks of a given board having three sides coincident with the boundary of the board is called a *knob*.

Lemma 1: [2] Let S be a maximal square in a simply-connected board B. Then either S is a disconnecting square or S contains a knob of B.

Theorem 1: Let B be a simply-connected board. Then the associated graph $G(B)$ is chordal.

Proof: By induction on the number of blocks in a board. As the induction hypothesis, assume the claim is true for simply-connected boards having fewer than m blocks. Let B be a simply-connected board having m blocks. Let S be a maximal square in B, and let $C = \{c_1, c_2, \cdots, c_n\}, n \geq 4$, be the set of blocks in B corresponding to the nodes of a simple cycle of length at least 4 in $G(B)$. By Lemma 1, there are two cases to consider.

Case I: S is disconnecting. Let $B_1, \ldots, B_j$, $j \geq 2$, be the components of $B-S$.

Observation 1: Two blocks are in a square of B iff they are in a square of $S \cup B_i$ for some $i \leq j$. (Because S is disconnecting.)

Suppose the blocks of C are all contained in $S \cup B_i$ for some i. Then by Observation 1, the subgraphs induced by C in $G(B)$ and in $G(S \cup B_i)$ are identical. Since the board $S \cup B_i$ has fewer than m blocks and is simply-connected, apply the induction hypothesis and conclude that C has a chord in $G(S \cup B_i)$, and thus also in $G(B)$.

Otherwise, suppose that some blocks of C are in different components of $B-S$, say B_p and B_q. By Observation 1, any path from B_p to B_q must pass through a block in the square S. Since C is a cycle, there are two distinct paths from any of its blocks in B_p to any of its blocks in B_q. Since C is vertex-simple, there must be two distinct blocks of C in S. The nodes in $G(B)$ corresponding to these two blocks in S are connected by an edge, and thus the cycle must have a chord.

Case II: S is not disconnecting. Then, by Lemma 1, S contains a knob of B. Call the knob K.

Observation 2: Any block in a knob contained in S can only be connected (in $G(B)$) to some other block in S.

If some block in C, call it c_i, is contained in any knob in S, then c_i is connected to two other blocks in S. These blocks are connected, so C has a chord.

Suppose no block of C is contained in any knob in the square S. Assume that K is a $1 \times p$ knob in a $p \times p$ square S, and that the board is oriented as shown in Figure 1. Then there are two cases to consider: either (A) each block in K has at least p blocks of B beneath it, or (B) some block of K, say b_1, has exactly $p-1$ blocks beneath it.

Case A: Each block in K has at least p blocks beneath it.

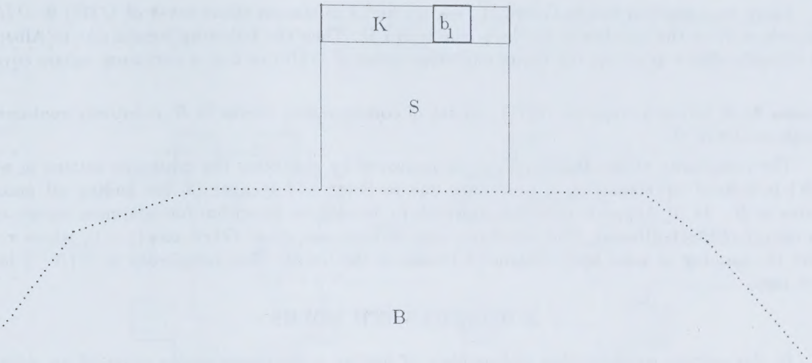


Figure 1

Consider the board $B-K$. By Observation 2, any block in K can only be connected to some other block in S. Thus, except for S, every square of B is also a square of $B-K$. So if two blocks of C are connected by a square of B other than S, they are connected by that same square in $B-K$.

Now suppose two blocks of C are connected by S in B. Since these blocks are not in K, they are connected by the $p \times p$ square consisting of rows 2 through p of S and the p blocks immediately below row p of S.

So two blocks of C that are not in K are in a square of B iff they are in a square of $B-K$. Thus the subgraphs induced by C in $G(B)$ and in $G(B-K)$ are identical. Since $B-K$ is simply connected and has fewer than m blocks, the induction hypothesis may be applied to conclude that C has a chord in $G(B-K)$ and thus also in $G(B)$.

Case B: Some block of K, call it b_1, has exactly $p-1$ blocks of B beneath it. The last of these must be a boundary block, call it b_2. See Figure 2.

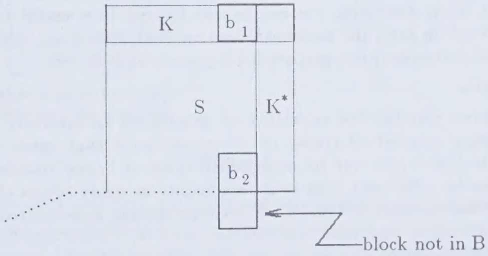


Figure 2

Since S is not disconnecting, one of the paths of border blocks of S from b_1 to b_2 must consist entirely of boundary blocks. Thus one of the vertical sides of S must be the boundary of another knob, call it K^*, as shown in Figure 2.

Let $L = K \cup K^*$. Consider the board $B-L$. Two blocks of C not in L are in a square of B iff they are in a square of $B-L$. The board $B-L$ is simply-connected and has fewer than m blocks, so the induction hypothesis may be applied to conclude that C has a chord in $G(B-L)$ and thus C also has a chord in $G(B)$. □

Using an algorithm due to Gavril [4], we can find a minimum clique cover of $G(B)$ in $O(N^{2.5})$ time, where N is the number of blocks in the board B. Then the following lemma due to Albertson and O'Keefe allows us to use the minimum clique cover of $G(B)$ to find a minimum square cover of B.

Lemma 2: [2] Given a clique in $G(B)$, the set of corresponding blocks in B is entirely contained in a single square of B.

The complexity of the algorithm can be improved by exploiting the geometric setting in which $G(B)$ is defined by employing a procedure due to Scott and Iyengar [9] for finding all maximal squares in B. In [1] Aupperle uses this approach to develop an algorithm for minimum square cover of a simply-connected board. The algorithm runs in time and space $O(rc\ min(r,c))$, where r and c are the number of rows and columns of blocks in the board. This complexity is $O(N^{1.5})$ in the worst case.

3. BOARDS WITH HOLES

In this section we show that the problem of finding a minimum square cover of an arbitrary board containing holes is an NP-complete problem. For a discussion of the theory of NP-completeness see [5].

Our proof uses a transformation from the NP-complete problem of Planar 3-Satisfiability (P3SAT). An instance of P3SAT is an instance of 3-Satisfiability with the additional property that a graph associated with the instance is a planar graph. The associated graph consists of a set of nodes corresponding to the variables and clauses, and with an edge between any variable and any clause in which that variable (or its negation) appears. See [6] for a more detailed discussion of P3SAT.

The transformation involves constructing an orthogonal polygon that simulates an arbitrary instance of P3SAT. The polygon must have the property that calculation of the size of a minimum square cover of the polygon will answer the question of satisfiability of the corresponding instance of P3SAT.

The polygon is built up from three types of components: variable loops, wires, and junctions. Variable loops allow the generation of an arbitrary number of copies of a variable and an arbitrary number of copies of the negation of that same variable. Wires allow the propagation of truth values from variable loops to junction figures. Junctions represent the logical AND of the three input values. The individual components are described in more detail below.

The orthogonal polygon is assembled from component pieces laid out on a grid of unit squares. For reference purposes when describing the component figures, it is useful to label alternating grid lines as "odd" and "even" in both the horizontal and vertical directions. All the squares needed for a minimum cover of the orthogonal polygon are 2×2 squares of 4 blocks.

3.1 Variable Loop Figures

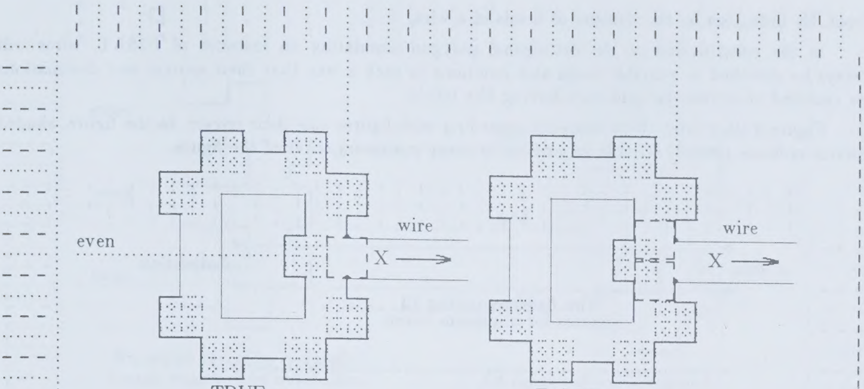
Variable loop figures provide the capability of generating an arbitrary number of copies of a variable and an arbitrary number of copies of the negation of that same variable. Variable loop figures are positioned so that wires may be attached to them at "even"-labeled vertical grid lines. A variable loop figure may be arbitrarily high to accommodate as many copies of a variable or its negation as are required by the instance of P3SAT. Wires representing non-negated variables are attached to the variable loop in such a way that they are centered on "even"-labeled horizontal grid lines. Wires representing negated variables are centered on "odd"-labeled horizontal grid lines.

A variable loop figure has two minimum covers. In one of these, squares protrude out from the variable loop figure and into the attached wire figures that represent non-negated variables. This cover corresponds to the assignment of TRUE to that variable. In the other cover, squares are flush with wires representing non-negated copies of the variable, and protrude out into the wires representing negated copies of the variable. This cover corresponds to the assignment of FALSE to that variable.

Figure 3 illustrates these concepts regarding variable loops and their covers. In the figure, shaded squares indicate those that must be present in any minimum cover. Dotted lines represent "even"-labeled grid lines; dashed lines represent "odd"-labeled grid lines.

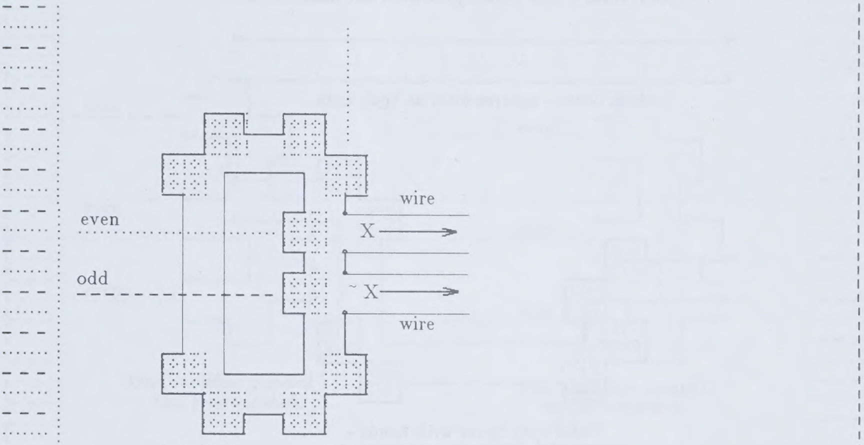
3.2 Wire Figures

A straight wire figure consists of a $2\times n$ rectangle, where n is the number of unit squares traversed by the figure in a horizontal direction, and is always an even number. Wires are always attached to variable loops and junctions at "even"-labeled vertical grid lines. A wire is considered to



TRUE cover. FALSE cover.

Variable loop generating a single variable.



Variable loop generating a variable and its negation.

Figure 3

be carrying a TRUE value when the first 2×1 rectangle on its left hand side (*source*) is covered by a square that protrudes out from a variable loop. Consequently, in a minimum cover, the right hand side of the wire (*destination*) has half a square protruding out from it and into a junction figure.

Wires can be bent at 90 ° angles as shown in Figure 4. When a wire is bent, it may no longer possess the desired behavior with respect to minimum coverage by squares. We say that a wire figure is *valid* with respect to propagation of truth values if it has exactly two minimum covers: one with squares flush at both the source and destination of the wire, and the other with half a square protruding from the destination of the wire into the attached junction figure. (In the latter case, coverage of the first 2×1 rectangular area at the source of the wire is provided by a square that protrudes from the attached variable loop.) The following lemma guarantees that the wires used in constructing the orthogonal polygon are valid.

Lemma 3: If a wire figure has its source and destination centered on horizontal grid lines having like labels (both "even" or both "odd"), then that figure is valid with respect to propagation of truth values.

Proof: By induction on the number of bends in a wire. □

In the construction of the orthogonal polygon simulating an instance of P3SAT, wires will always be attached to variable loops and junctions in such a way that their sources and destinations are centered on horizontal grid lines having like labels.

Figure 4 illustrates these concepts regarding wire figures and their covers. In the figure, shaded squares indicate those that must be present in every minimum cover of the figure.

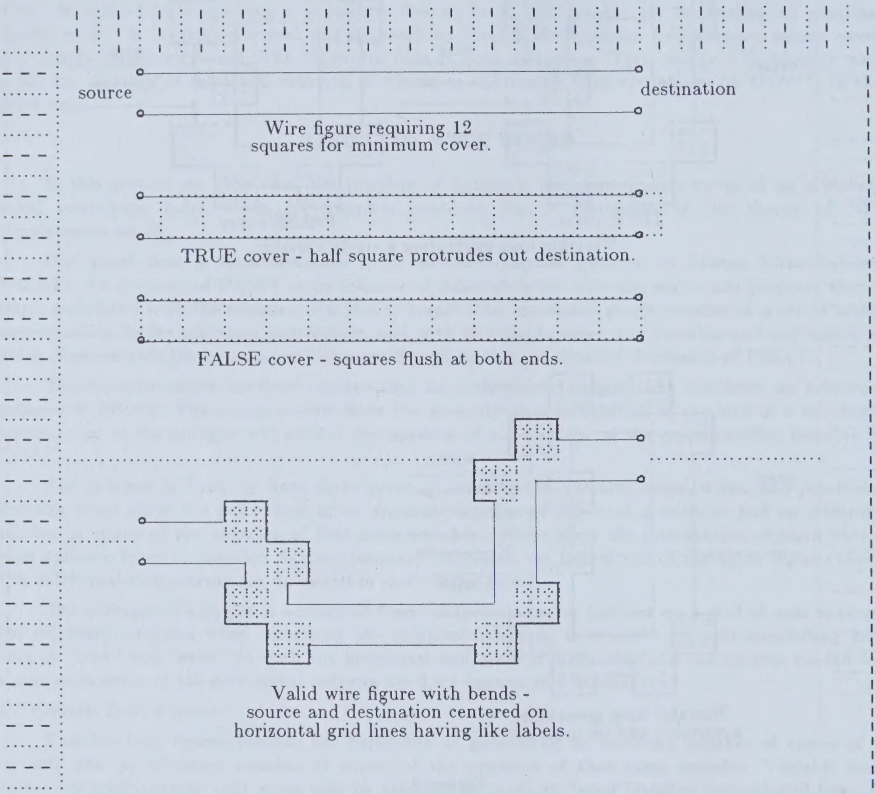

source
destination
Wire figure requiring 12
squares for minimum cover.
TRUE cover - half square protrudes out destination.
FALSE cover - squares flush at both ends.
Valid wire figure with bends -
source and destination centered on
horizontal grid lines having like labels.

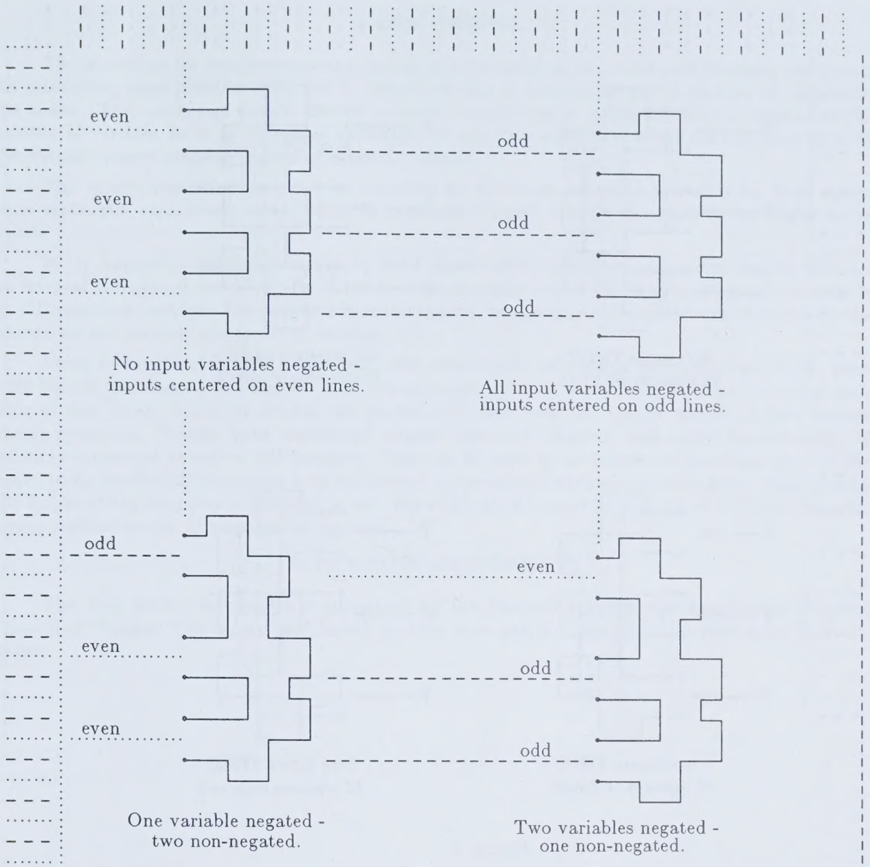
Figure 4

3.3 Junctions

Junction figures provide the capability of testing the value of the logical AND of the three input variables. They are positioned with their left boundary aligned on an "even"-labeled vertical grid line. Input wires representing non-negated variables are centered on "even"-labeled horizontal grid lines, and input wires representing negated variables are centered on "odd"-labeled horizontal grid lines. Two slightly different junction figures are used: one for clauses in which all the variables are either negated or non-negated, and one for clauses in which there are two negated variables and one non-negated or vice versa. Since the order of appearance of the literals (variable or negation of a variable) in a clause does not affect the truth value of the clause, assume that the odd literal appears first. Figure 5 displays the two types of junction figures and the various combinations of inputs.

Lemma 4: A junction figure can be covered with 12 squares iff all three input wires carry a value of TRUE . If any input wire carries a value of FALSE , then 13 squares are required to cover the junction.

Proof: Omitted.



No input variables negated - inputs centered on even lines.

All input variables negated - inputs centered on odd lines.

One variable negated - two non-negated.

Two variables negated - one non-negated.

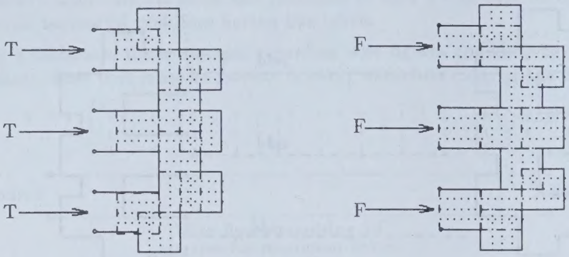
Figure 5

Figure 6 displays minimum covers of a junction figure for representative combinations of input values. Covers of the other type of junction figure and other combinations of inputs are similar.

An instance of P3SAT consists of the conjunction of a set of clauses, where each clause is the disjunction of three literals. In testing the satisfiability of such an instance, it would be useful to have a junction figure that simulates the logical OR of three input literals. Such a junction figure is presented in [1]. However, the AND junction figure described here is somewhat simpler, and can also be used to test the satisfiability of an instance by applying DeMorgan's law as follows. Negate each literal in the instance and let each OR connective in each clause be replaced by an AND connective. Now if every assignment of truth values to the variables in the instance results in every clause having value FALSE , then this is equivalent to each of the original clauses being made TRUE , and the instance is satisfiable.

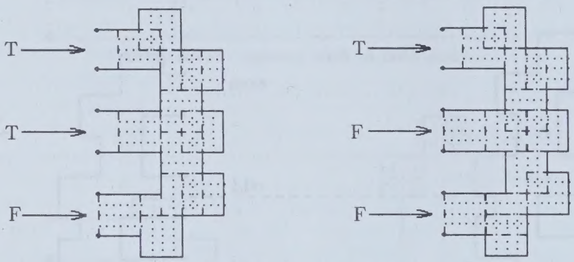
3.4 Overview of Construction

Given an instance I of P3SAT, examine the literals in each clause. If any clause consists of one negated variable and two non-negated variables, rearrange the order of appearance of the literals (if necessary) so that the negated variable appears first in the clause. If any clause consists of two negated variables and one non-negated variable, rearrange the order so that the non-negated variable appears first. After this rearrangement, every clause has its literals in an order that is compatible



All inputs TRUE - 12 squares required.

All inputs FALSE - 13 squares required.



Two inputs TRUE - 13 squares required.

One input TRUE - 13 squares required.

Figure 6

with one of the two junction types. The rearrangement process does not affect the satisfiability or the planarity of the instance of P3SAT.

Next apply DeMorgan's law as previously described to convert OR connectives into AND connectives. During this process all literals will be negated.

For every variable, scan the clauses counting all appearances of that variable and all appearances of its negation. Set up a variable loop with the required number of copies of the variable and its negation. Count the number of squares required to cover all such variable loops. Let this number be V.

Run wires from the variable loops to junction figures representing the clauses. Count the number of squares required to cover the wires. Let this number be W, and let the number of junction figures be J.

The preceding construction can be done in polynomial time. Now, find a minimum square cover of the orthogonal polygon just constructed. Let the number of squares in the minimum cover be C. If $C = V + W + 13 * J$, then I is satisfiable. If $C < V + W + 13 * J$, then I is not satisfiable.

Thus if the minimum square cover could be found in polynomial time, we would also have determined the satisfiability of I in polynomial time, so finding the minimum square cover must be NP-hard. Clearly the minimum square cover problem is in NP, so we have the following theorem:

Theorem 2: The problem "are there k squares that cover an arbitrary orthogonal polygon?" is NP-complete.

4. DISCUSSION

The algorithm for simply-connected boards is polynomial in the number of blocks in the board. An interesting open problem is to find an algorithm that is polynomial in the number of vertices in the board. The number of squares needed to cover a board can be arbitrarily high in relation to the number of vertices, as in an elongated rectangle. So any such algorithm would have to have some of the output squares encoded instead of explicitly output.

The results presented here can be extended to minimum coverage of boards by fixed aspect ratio rectangles of a given aspect ratio. A rectangle's aspect ratio is the ratio of its height to its width.

In [1] Aupperle considers coverage by fixed aspect ratio rectangles having side lengths that are an integral multiple of one block. He shows that for multiply-connected boards, minimum coverage is an NP-complete problem. For simply-connected boards, he shows that the graphs associated with the boards are not perfect, and thus not chordal.

Using rectangles of a fixed aspect ratio with side lengths an integral multiple of one block, there exist boards that cannot be covered. Another approach is to allow side lengths to be a rational multiple of one block. Then all boards can be covered, and both the results presented here extend: simply-connected boards have associated graphs that are chordal, and minimum coverage of multiply-connected boards is NP-complete. This can be seen by performing a transformation on the board to be covered. If the board is to be covered by rectangles with aspect ratio h/w, then increase the height of the board by a factor of w and the width of the board by a factor of h. Then the problem is reduced to one of coverage by squares.

5. ACKNOWLEDGEMENTS

The first author was partially supported by the Natural Sciences and Engineering Research Council of Canada. The second and fourth authors were partially supported by NSF grant DCR-83-51468.

REFERENCES

[1] L. J. Aupperle, "Covering Regions by Squares," Master's Thesis, University of Saskatchewan, Saskatoon, Canada, July 1987.

[2] M. O. Albertson and C. J. O'Keefe, "Covering regions with squares," *SIAM J. Alg. Disc. Meth.*, vol. 2, 3, pp. 240-243, Sept. 1981.

[3] J. C. Culberson and R. A. Reckhow, "Covering Polygons is Hard Preliminary Abstract," Tech. Rept. TR 88-6, Department of Computing Science, University of Alberta, 1988.

[4] F. Gavril, "Algorithms for minimum coloring, maximum clique, minimum covering by cliques and maximum independent set of a chordal graph," *SIAM J. Computing*, vol. 1, 2, pp. 180-187, June 1972.

[5] M. R. Garey and D. S. Johnson, *COMPUTERS AND INTRACTABILITY A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, New York, 1979.

[6] D. Lichtenstein, "Planar formulae and their uses," *SIAM J. Comput.*, vol. 11, 2, pp. 329-343, May 1982.

[7] J. O'Rourke, *Art Gallery Theorems and Algorithms*, Oxford University Press, New York, 1987.

[8] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, Academic Press, New York, 1982.

[9] D. S. Scott and S. S. Iyengar, "TID - a translation invariant data structure for storing images," *Communications of the ACM*, vol. 29, no. 5, pp. 418-429, 1986.

[10] A. Y. Wu, S. K. Bhaskar, and A. Rosenfeld, "Computation of Geometric Properties from the Medial Axis Transform in $O(n \log n)$ Time," *Computer Vision, Graphics, and Image Processing*, v. 34, 76-92, 1986.

AN EFFICIENT ON-LINE LINEAR CHEBYSHEV APPROXIMATION ALGORITHM FOR A FINITE PLANAR POINT SET

BINAY K. BHATTACHARYA
Simon Fraser University, Burnaby, BC, Canada, V5A 1S6

Abstract

In this note we have presented an on-line linear Chebyshev approximation algorithm for n planar point set. The algorithm takes O(logn) time per point insertion. This improves the O($\log^2$n) time algorithm of Rey and Ward [RW]. A data structure has been proposed which allows efficient implementation of operations of various kinds on convex polygons.

1. Introduction

Given data points $p_i = (x_i, y_i)$, i=1,2,...,n and a line described by y = ax + b, define

$$e_{a,b} = \max |y_i - (ax_i + b)|. \qquad (1)$$

The above line is minimax linear-fit if it minimizes $e_{a,b}$. The line is also known as Chebyshev approximant to the given data points.

Many researchers have looked at the off-line version of this problem (see [RW] for the complete list). Shamos [S] provided an O(nlogn) geometric algorithm to solve the linear Chebyshev approximation problem. Megiddo [M] noted that the Chebyshev approximation could be expressed as a linear programming problem in 3-dimension and therefore can be solved in linear time. This note deals with updating the linear Chebyshev approximation of a finite point set. Given the linear Chebyshev approximation of a point set S, how one can efficiently compute the linear Chebyshev approximation of S $\cup$ {p} for some new point p (point insertion), or how one can compute the linear Chebyshev approximation of S - {p}, where p is a point of S (point deletion). Rey and Ward [RW] designed an O($\log^2$n) solution for the insertion problem which is based on the algorithm of Shamos [S]. In this note we present a simple O(logn) algorithm for the point insertion case.

2. Geometric properties

Let S be a set of points in the plane and let CH(S) denote its convex hull. We state the following results which are already known.

Theorem 1: [R]A line is the Chebyshev approximant of S in the plane if and only if it maximizes the y-distance of a best Chebyshev approximation among all subsets consisting of three points. ☐

The set of points satisfying Theorem 1 will be known as Chebyshev points.

Theorem 2: [S]There exists a set of Chebyshev points of a finite set that are vertices of its convex hull and two of these points are consecutive on the hull. ☐

Let e be an hull edge and let v be the vertex through which the supporting line to S, parallel to e, passes. We will call v as the **opposing vertex** of e and e as an **opposing edge** of v. We will also call e and v as **antipodal pair** and will be denoted as (e,v). It can easily be shown that

Theorem 3: If (e,v) is an antipodal pair of CH(S), v maximizes the y-distance of the points of CH(S) to e. ☐

Let e_i and e_j be two non-adjacent edges of CH(S) and suppose there exists a vertex v such that (e_i,v) and (e_j,v) are antipodal pairs. Let C(i,j) denote the convex chain of CH(S) in anti-clockwise order where e_i and e_j are respectively the first and the last edge. v can either belong to C(i,j) or C(j,i). Without any loss of generality assume that no edge in C(i,j) is incident on v. Let e_k be an edge in C(i,j), k≠i,j. It is easy to show that

Lemma 1: (e_k,v) is also an antipodal pair. ☐

Lemma 1 implies that each edge of C(i,j) forms an antipodal pair with v. Let e_k, k=i, ..., j, be the only edges of P which form antipodal pair with v. Let L_k be the line through e_k, k=i, ..., j. Let L be the vertical line through v. Let q_k be the intersection point of L_k with L. If L_k and L are parallel, q_k is a point at infinity. Let q_m be the intersection point nearest to v. It can be shown that

Theorem 4: The vertical distance of v to e_m is the minimum among the vertical distances of v to the edges of C(i,j). ☐

This vertical distance will be called **minimum y-distance** of v.

3. Proposed algorithm

It is assumed that the point p_i arrives after p_{i-1}, i=2,3,...,n. Let CH(S_i) be the convex hull of the set $S_i = \{p_1, p_2, ..., p_i\}$. It is also assumed that CH(S_i) is known and the point p_{i+1} is being processed. Let e_j, j=1,2,...t, be the edges of CH(S_{i+1}) in anti-clockwise order where e_1 and e_t are incident on p_{i+1}. Let q_i, i=1,2,...,t be the vertices of CH(S_{i+1}) in anti-clockwise order with $q_t \equiv p_{i+1}$.

The algorithm **CHEBYSHEV** is formally described below:

```
Step 1: If $p_{i+1}$ lies inside CH($S_i$) then exit.
            {In this case the linear Chebyshev approximation is the same as that of the points $p_j$, j=1,2,...,i.}
Step 2: {$p_{i+1}$ lies outside CH($S_i$)}
            Update CH($S_i$) to obtain CH($S_{i+1}$).
```

Step 3: Determine the opposite vertices q_j and $q_{j'}$ of e_1 and e_t respectively.
Step 4: Determine the opposing edge list of q_k, k=1, j′, j, t-1, t.
Step 5: Determine the minimum y-distance of q_k, k=1, j′, j, t-1, t.
Step 6: Determine the minimum of the minimum y-distances of q_k, k=1,2, ..., t.

3.1 An implementation

We assume that the vertices of CH(S_i) are traversed in anti-clockwise direction. The cyclic ordering of the vertices of CH(S_i) induces a direction for the edges of CH(S_i). We store these directed edges in order in the leaf nodes of a 2-3 tree [AHU]. Each directed edge is represented by a 2-tuple of the form (p,q) where p and q are the vertices of CH(S_i) determining the edge and the direction of the edge is from p to q. The directed edge with the smallest angle with the +ve x direction is stored in the leftmost leaf node. We store at each internal node v the number of leaf nodes in the subtree with v as the root. Also at v we store two pointers which points to the leftmost and the rightmost leaf nodes of the subtree with v as the root. This structure is somewhat similar to the one proposed in [AES]. This data structure allows us to do each of the following operations in time proportional to the height of the tree.

(i) Determine whether p_{i+1} lies inside CH(S_i),
(ii) Update CH(S_i) to obtain CH(S_{i+1}),
(iii) Given a direction determine the pair of points of CH(S_i) which support lines that are parallel to the given direction,
(iv) Given a line determine the intersection points of the line with CH(S_i), if exist, and
(v) Given a vertex of CH(S_i) determine its opposing edges.

Using the results of Section 2 we notice that steps 1 through 5 of CHEBYSHEV can be implemented in O(logn) time. Thus we are left with the problem of implementing step 6.

We assume that the minimum y-distance values of the vertices of CH(S_i) are known. A vertex which is not the opposing vertex of any edge is considered to realize minimum y-distance of value infinity. Only the vertices with finite minimum y-distances are stored in a 2-3 tree. Each leaf node stores a vertex, its minimum y-distance and the corresponding opposing edge that realizes this y-distance. These vertices are stored in increasing polar angles where the polar angle of a vertex is the angle at the origin that this vertex makes with the positive x direction. The origin of the coordinate system is assumed to be inside CH(S_i). We now formally describe the various sub-steps of step 6.

(a) Delete from the 2-3 tree the nodes representing the vertices of CH(S_i) which are no longer the vertices of CH(S_{i+1}).
(b) Delete from the 2-3 tree the nodes representing the vertices of CH(S_{i+1}) which lie on the opposing edges of p_{i+1}.
(c) Delete from the 2-3 tree any node that represents q_k, k=1, j′, j, t-1.
(d) Insert a node for the vertex q_k, k=1, j′, j, t-1, t.
(e) Determine the minimum y-distance.

Since the vertices are stored in a 2-3 tree, the steps (a) through (e) can be implemented in O(logn) time.

Thus we see that the total time required to report the Chebyshev points when p_{i+1} is inserted is O(logn).

REFERENCE

[AES] Avis,D.; ElGindy, H. and Seidel, R., "Simple on-line algorithms for convex polygons," In Computational Geometry, G.T. Toussaint (Editor), North Holland, pp.23-42, 1985.

[AHU] Aho, V.A.; Hopcroft, J.E. and Ullman, J.D., The design and analysis of computer algorithms, Addison Wesley, 1974.

[C] Coxeter, H.S.M., "The classification of zonohedra by means of projective diagram," In Twelve Geometric Essays, Southern Illinois University Press, pp.137-156, 1968.

[M] Megiddo, N., "Linear programming in linear time when the dimension is fixed," Journal of ACM, Vol.31, No.1, pp. 114-127,1984.

[R] Rice, J.R., Linear Theory: The approximation of functions, Vol.1, Addison Wesley, 1964.

[RW] Rey, C. and Ward, R., "On determining the on-line minimax linear fit to a discrete point set in the plane," Information Processing Letters, Vol.24, pp.97-101, 1987.

[S] Shamos, M.I., Computational geometry, Ph.D., Thesis, Yale University, 1978.

DESIGN AND ANALYSIS OF A HIERARCHICAL SNOOPING CACHE COHERENCE SYSTEM

MANI AZIMI AND CARL ERICKSON
Department of Electrical Engineering
Michigan State University
East Lansing, Michigan 48824

ABSTRACT

A new hierarchical cache coherence protocol is proposed to maintain cache coherence in hierarchical bus-based shared-memory multiprocessors. Coherence maintenance in the distributed cache memory designs of such multiprocessors plays a crucial role in the overall performance of the system. The protocol is designed to allow efficient sharing of cache blocks across the system, while adding minimal excess load on system buses. Compatibility with Futurebus and modest hardware requirements of this design allow its implementation in future high performance multiprocessor systems. Modeling and performance evaluation of the protocol by means of colored stochastic Petri nets is currently underway by the authors.

INTRODUCTION

Single-bus shared memory multiprocessors are constrained by an upper limit on the number of processors, mainly due to the saturation of the system bus [1]. One way to overcome the bus saturation problem is to group the processors in clusters and thus create a hierarchy. Hierarchical bus-based systems provide plausible solutions for many applications. One can envision a single high speed bus or an interconnection network to connect several clusters of processors. With the interconnection network approach, if the number of clusters is much smaller than the number of processors, the complexity of the required interconnection network may be tolerable for some systems. In this paper, however, we concentrate on the global bus approach, in which clusters of processors are connected with a single high speed bus. In such an architecture, interprocessor and processor-memory communications are more costly when the two participating parties are not located on the same cluster bus. With support from compilers and intelligent task schedulers in the operating system, a wide range of applications can be run efficiently on the proposed architecture.

The processor-memory interconnection of the shared memory multiprocessor of interest is illustrated in Figure 1. The detailed structure of each cluster in Figure 1a is shown in Figure 1b. The processors within a cluster are connected via the cluster buses. A single global bus interconnects the clusters and is the only means of communication among clusters. The memory modules are distributed throughout the system to minimize the overall memory access costs of the system and make intracluster activities independent of the global bus. It is assumed that processor assignment and job scheduling are performed with the knowledge of the underlying architecture. The global bus memory module can be used for holding global data which is widely shared between clusters. Shared I/O devices are connected to the global bus so that all processors can have access to them in a balanced way, while individual clusters might have their own disk I/O.

The focus of this paper is on the mechanisms of maintaining cache coherence within the shared memory multiprocessor shown in figure 1. An efficient distributed cache memory design and its corresponding data coherence protocol play a major role in boosting the performance of these systems. We will show that, utilizing reasonable hardware and commercial buses available over the next few years [2], one can efficiently maintain cache coherence in such a system with the proposed cache coherence protocol. Well-designed private cache memories not only improve memory access time, but also reduce the traffic through the interconnection network.

In general, data incoherence results whenever some changes to the data are not globally observed in the system (i.e. by all holders of the data). This issue is of significance in shared memory multiprocessors where the crucial goal is data sharing. Our design objectives for the cache coherence maintenance system are three-fold. We seek to minimize: 1) the excess access time for nonshared data added by the coherency protocol, 2) the shared data access time, and 3) the protocol traffic load on system buses.

Wilson has recently designed a two-level cache hierarchy for a cluster based multiprocessor in which clusters of processors are connected by means of a global bus [3]. A cluster bus connects the processors within each cluster together. In this design both cluster and global buses run the same protocol for maintaining cache coherence. The protocol Wilson describes is basically a write-invalidation type. The second level caching is performed at the interface of the two type buses. The cluster cache controller acts as a cache device for that cluster, and it holds a copy of every processor cache entry on that cluster. In Wilson's estimation, the size of this cluster cache controller must be ten times the sum of the individual processor caches on the cluster for which it is responsible. Baer and Wang [4] have recently studied the inclusion problems associated with multi-level caching in general, and in the proposed system by Wilson in particular.

In this paper, we are proposing a cache coherence system for a two level hierarchical bus-based architecture. Cache Controllers (CC) maintain the coherency of the shared blocks on the behalf of their corresponding processors. Cluster Memory Controllers (CMC) on different clusters create the illusion of a global homogeneous local address space to the processors of the same cluster. Cluster Cache Controllers (CCC) maintain the coherency of those blocks cached on their clusters and shared with other clusters.

Each cluster bus runs a modified Dragon protocol for cache coherency maintenance. The distinguishing feature of this write-back protocol is the sharing of dirty blocks between processors without constant memory intervention. This allows for efficient sharing of data within a cluster. Blocks may be "checked out" from memory for writing, and shared among processors without being written back to memory until purged. In the applications suitable to the architecture under consideration, the majority of processor memory accesses are destined to local memory modules. For this reason, efficient support of intra-cluster sharing is of paramount importance in the cache coherence protocol.

The CMCs support distributed memory modules by providing transparent access to remote memory locations for the processors. The cache coherency protocol on the global bus is of the simple write-back-invalidate type. This results in reasonable intercluster sharing costs and the protocol traffic on the global bus is reduced. Sharing of data for writing purposes among clusters is not allowed; instead a series of read/invalidate/write actions takes place. Processors must first secure a copy of a block for reading, then upon writing to this block, invalidate copies in other clusters. For performance reasons, the majority of memory accesses will be to locations local to the home cluster of a processor. However, the CMCs, by providing transparent support of a single memory space, simplify the task of programming such a multiprocessor.

The main advantage of our system over Wilson's is that instead of holding copies of all blocks cached in the cluster caches, our CCC holds only the *status* of those blocks. This reduces the required size of the CCC dramatically. The cost of this technique is the increase in the global bus traffic for *simultaneous* reads of the *same* remote blocks by distinct processors on the same cluster. If the CCC were to cache the remote blocks in it's cluster, then it could service additional read requests of the *same* remote block for that cluster. With our system, each new processor read must be serviced by the appropriate remote CMC generating a global bus read request. It is clear that this type of sharing in the architecture of interest is quite rare.

Archibald has quite recently published a protocol for a hierarchical cluster-based multiprocessor [5]. Surprisingly, his protocol, independently developed, is similar in some aspects of design to the one presented here. For instance, Archibald also takes the approach of caching only state information in cluster caches. One major advantage of the architecture adopted in this paper is that memory modules are distributed throughout the system, while the centralized global memory approach of Archibald inevitably results in global bus saturation. Archibald's more complicated coherence protocol has nine block states, requiring much more complex cache controllers, an important point that he seems to have ignored. Detailed modeling of the system presented here has given the authors a keen appreciation of the complexities involved in the controller design.

While our protocol is simple and efficient, as intercluster activities in a cluster based system exceed a threshold, the global bus becomes a bottleneck. In that case, the global bus has to be replaced with an interconnection network with a larger bandwidth. An avenue of future research is cache coherency on clusters joined by general interconnection networks, such as the Harris Concert multiprocessor [6]. The initial attempt of Scheurich and Dubois requires a locking mechanism to preserve the consistency [7]. It would be interesting to explore more efficient solutions for systems employing similar architectures.

DESIGN OF THE HIERARCHICAL CACHE COHERENCE PROTOCOL

The basic two level hierarchical structure of a bus-based shared-memory multiprocessor is shown in Figure 1. Processors within a cluster are connected via a cluster bus, while a single global bus interconnects the clusters and is the only means of communication among clusters. The memory modules are distributed throughout the system to minimize the overall memory access costs across the system and isolate intracluster activities from the global bus. The memory unit on the global bus can hold the global data shared among clusters for efficient accesses.

The units labeled *Cache Controller (CC)* respond to their local processor read and write requests. Cache controllers also maintain coherence among the processor caches on the same cluster. The CCs communicate via a Dragon-like protocol [8] to optimize intracluster sharing of blocks for both reading and writing purposes.

The blocks labeled *Cluster Cache Controller (CCC)* monitor both their respective cluster bus and the global bus. These controllers read addresses and associated commands off the transactions on both buses and take appropriate action according to the state of the referenced block in their directories. The main function of a CCC is to isolate the local traffic on its cluster bus from the general traffic on the global bus. A CCC is informed of the cleanness and shared status of the blocks present in its local caches (caches existing on the same cluster as the CCC itself), to be able to actively participate in the related global bus transactions. We emphasize that CCCs do not hold the actual cached data, but cache only the *status* of those blocks. When a CCC must respond to a global bus transaction by taking some action on its cluster bus, it preempts any existing local cluster bus transaction requiring the global bus. By preempting those transactions CCCs avoid deadlock.

The blocks labeled *Cluster Memory Controller (CMC)*, placed at the interface of cluster buses and the global bus, present as much as possible, a unified image of the global memory to devices on that particular cluster bus. CMCs are responsible for responding to local cluster generated requests requiring access to remote memory modules. CMCs also control the response of their local memory modules and inhibit them from responding to requests for invalid remote blocks. To operate in compliance with the protocol, CMCs need to hold the status of those blocks belonging to their local cluster memory module.

Three major issues were considered in the design of the controllers and bus protocols. First, the global bus utilization in this hierarchical system should be constrained. The global bus can easily become the system bottleneck since it may handle much of the system I/O as well as the intercluster activities. Second, the delay associated with the intracluster activity should be minimized since the system is mainly designed for efficiency of intracluster operations. Third, the communication costs for intercluster accesses should be within an *acceptable* range to allow sharing of data amongst clusters.

To maintain efficient intracluster operations, the most advanced single bus protocol is chosen for the bus clusters. We have slightly modified the Dragon protocol for this purpose [8]. This minor modification forces each CC to broadcast its first write into any block entry in its cache, and thus inform its local CCC of the "dirtiness" of that block. The notification is required due to the monitoring responsibility of CCCs of the global bus transactions. This broadcast introduces a minimal load on the cluster bus and we expect only a minor effect on the overall protocol performance.

Blocks in CCs may be in any of the following states: INVALID (the cached block is stale), SHARED-CLEAN (the block may be shared by other processors and the copy in memory might be stale, where in that case the write-back is the responsibility of the local cache controller having the block in the DIRTY state), DIRTY (the local write into the block has not yet been reflected in the corresponding memory module and the CC has exclusive access to the block), and finally SHARED-DIRTY (the local write into the block has not yet been reflected in the corresponding memory module but the block may be shared with other local CCs).

To keep the global bus utilization low, a simple write-back plus invalidation scheme is used for intercluster activities. Protocols which require broadcast of writes to shared entries over this bus are excluded from consideration. Following such a protocol, CCCs invalidate remote copies of blocks after a local write to them, thus avoiding broadcast of write notices on the global bus. Although this decision discourages sharing of "dirty" copies among clusters, it has no effect on sharing of "clean" entries. In addition, CCCs are able to own local blocks (blocks belonging to the memory module on the same cluster) exclusively to avoid the necessity of broadcasting global bus invalidations for blocks not shared with other clusters.

Blocks in a CCC may be in any of the following states: INVALID (no local cache holds a copy), VALID-EXCLUSIVE (the block is shared for reading purposes among local caches and is owned by the local memory module), VALID (the block may be shared for reading purposes by local caches as well as possibly some remote clusters), and finally DIRTY (the corresponding memory module, which might be on a remote cluster, has a stale copy and one local cache is responsible for the write-back).

As we mentioned the role of the CMC unit is to present a unified image of the memory system to its local caches. While CCCs keep track of the blocks in their local caches, the CMCs perform the bookkeeping on the blocks in their corresponding memory modules. To support the state of VALID-EXCLUSIVE in the CCCs the CMCs should also inform them of the exclusiveness of the block to that cluster at appropriate times. To perform this duty the CMCs have to distinguish between those blocks checked out by remote clusters and those used exclusively in the local cluster. This suggests distinct states of VALID and VALID-EXCLUSIVE in the CMCs. Also to minimize the number of accesses to the global bus the CMCs should be aware of the shift of ownership of DIRTY blocks to remote clusters. For this reason, we have broken down the INVALID state into two distinct states in the CMCs. If a DIRTY block is owned remotely, upon receipt of a local read request, the CMC should pursue the write-back of the block over the global bus, while in the case of a local ownership no access to the global bus is required.

In supporting distributed memory modules, CMCs respond to local transactions concerning blocks in other memory modules. For this reason we include an additional state called REMOTE, which represents blocks owned by remote memory modules. Note that no state information is held for REMOTE blocks. The possible states of a block in a CMC are: REMOTE (the block is owned by a remote memory module), INVALID-LOCAL (a local cache has a dirty version of the block), INVALID-REMOTE (a remote cache has a dirty version of the block), VALID (the block is shared by one or more local caches as well as possible remote caches), and finally VALID-EXCLUSIVE (the block is shared for reading purposes only by local caches).

DETAILED DESCRIPTION OF THE PROTOCOL

In the following, we consider actions taken by CCs, CCCs, and CMCs in response to bus transactions observed, requests generated by a local processor, and purging operations. The global bus should support the following three types of transactions: GBRR (Global Bus Read Request), GBWB (Global Bus Write Back), and GBIN (Global Bus INvalidate). The following transactions are the only transactions required on cluster buses: CBRR (Cluster Bus Read Request), CBWN (Cluster Bus Write Notice), CBWB (Cluster Bus Write Back), CBIN (Cluster Bus INvalidate), CBFL (Cluster Bus FLush), and CBFI (Cluster Bus Flush and Invalidate). Processors send only two types of requests: PR (Processor Read) and PW (Processor Write). PURGE (block PURGE) is an operation initiated by CCs or CCCs related to their block replacement schemes.

We have used the abbreviation CSHL(Cluster SHared_Line) and REML(REMote_Line) for the extra signal lines required on cluster buses. Protocols based on the write broadcast require an extra signal line, called SHARED LINE, for the exchange of information regarding the shared status of blocks. After any broadcast of write into a block by a cache other caches holding the block are responsible to inform the owner of their sharing of the block. This is accomplished through raising of the SHARED LINE. Since this line is an ORed signal, there would be no problem in having more than one cache raising it. In the hierarchical protocol, in special situations, the CMCs also use this line to signal to CCCs of the sharing status of the block with other remote clusters.

Since the CCCs can not distinguish the locality or remoteness of addresses, the CMCs use a second bus line, called REML, to inform them of the local or remote nature of blocks in a read request.

The information concerning the state transitions of the controllers, as well as their responses to processor requests and buses transactions, are tabulated in Tables 1-6. Each row of the tables represents a state that a block can be in in that particular unit. The columns represent various bus transactions, processor requests, and purging actions. Each row is further divided into two; the top part contains the actions taken by the device while the bottom one illustrates the changes in the state of the block within the device. The notation "sigline? action1 : action2." is used to indicate that 'action1' is taken if 'sigline' is raised and The notation □ signifies that the related transactions can not occur for a block in that state.

We can easily show that our protocol is compatible with the Futurebus standard. There are already provisions made in the protocol for the implementation of the SHARED LINE. One can easily implement the REML using the extra lines in the Futurebus specification reserved for specific designs. The IEEE P896 bus standard (Futurebus) was designed to support multiprocessor systems built with current and future generations of powerful 32 bit microprocessors. The P896.2 Working Group has defined a class of compatible cache coherency protocols called the MOESI protocols [9]. Boards using MOESI compatible protocols should be able to operate simultaneously on the Futurebus [2].

To illustrate the protocol operation, as well as giving an example of how to interpret the concise notation used in Tables 1-6, we will discuss an involved scenario of intercluster cache references here. A cache miss on a P-1 read request causes CC-1 to broadcast a read request (CBRR) over CBUS-1 (see Figure 2). Assuming that the block is owned locally and no other cache has a copy, M-1 responds to the request and provides the block. CC-1 reads the block off CBUS-1 and sets its state to SHARED-CLEAN. CCC-1 observes the read request on CBUS-1 and checks the REML to make sure that the request is not generated remotely and relayed by CMC-1 over CBUS-1. Next it watches for CMC-1 to raise the CSHL line to see whether the block is already being shared by other clusters. In the absence of both of these signals CCC-1 accommodates space for the block state and sets it to VALID-EXCLUSIVE, meaning that it is owned locally and no remote cluster has a copy.

A subsequent read request by P-10 and a CBRR broadcast over CBUS-1 by CC-10 is answered again by M-1. CC-10 loads the block off the bus and sets its state to SHARED-CLEAN. Now let's say P-10 writes into the block and thus CC-10 broadcasts the written word over CBUS-1 (CBWN) and changes the state to SHARED-DIRTY. CC-1 upon observing the word broadcast (CBWN) updates that particular word in its block, but keeps the block in the state SHARED-CLEAN and raises the CSHL, telling CC-10 to retain the block SHARED-DIRTY. CCC-1 upon observing the modification to the block changes the state of the block to DIRTY, while CMC-1 changes it to INV-LOC, indicating that some local cache has the last updated version.

Let's assume at this point P-11 performs a read causing a read miss (see Figure 3). Due to the space limitation in Figure 3, the global bus is shown in two sections. To get the block, CC-11 generates a read request over CBUS2. CMC-2 being oblivious to the state of the block in caches on cluster 2 has to wait and listen to the CSHL on CBUS-2 to make sure that no local cache is trying to respond to the read request. In this case, due to the lack of such response, CMC-2 broadcasts the request over the global bus. CCC-1, having the block in the state DIRTY, generates a flush request on CBUS-1 (CBFL) and changes its state to VALID, indicating that although the block is being written back it will still be used by local caches for reading purposes. M-1 observes the write-back (CBWB) and in the absence of any inhibition by local controllers, updates its entry. CC-10, having its state in the SHARED-DIRTY state, responds to the flushing request and writes the block over CBUS-1 (CBWB) and changes the state to SHARED-CLEAN. CC-1 does not react to the flushing request.

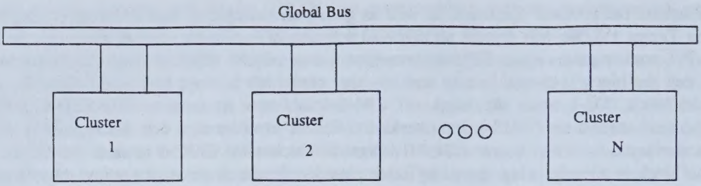
In the mean time, CMC-1 having the block in the INV-LOC state has recognized the address on the global bus and anticipates the subsequent flush request on CBUS-1. It simply reads the block off CBUS-1 and relays it over the global bus (GBWB).

CMC-2 reads the block off the global bus and relays it to CBUS2. It also raises the CSHL to signal to CCC-2 that the block is owned remotely and might be shared with them as well. In the mean time, CCC-2 has provided space for the block state. Upon observing the CSHL signal from CMC-2 it sets the block state to VALID, indicating that the block is shared with other clusters and/or owned remotely. Finally CC-11 reads the block off CBUS-2 and sets its state to SHARED-CLEAN.

Now let's consider a write to the block of interest by P-11. CC-11 broadcasts the written word over CBUS-2 (CBWN), listens to the CSHL line and due to lack of CSHL signal changes the status to DIRTY. CCC-2 observes the word write on CBUS2, broadcasts an invalidation signal on the global bus (GBIN), and changes its state to DIRTY. CCC-1 observes the invalidate signal, relays it over to CBUS-1 (CBIN), and sets the state of the block to INVALID. CC-1 and CC-10 invalidate their copies upon observing the invalidate signal on CBUS-1. CMC-1 changes the state to INV-REM upon observing the invalidate signal on the global bus, meaning that a remote cache has the last updated version.

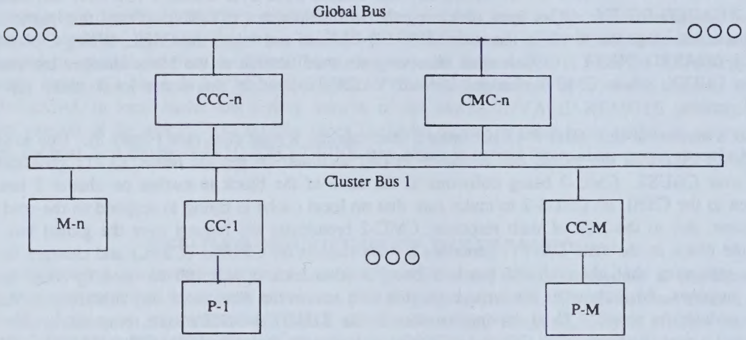
MODELING AND SIMULATION OF THE PROTOCOL

The design and evaluation of cache coherency protocols for hierarchical bus shared memory multiprocessors is a complicated endeavour. Modeling with High Level Timed Petri nets (HLTPN) has allowed us to capture the details of a hierarchical bus multiprocessor, and represent precisely the behaviour of the cache coherence protocol. With a HLTPN we can model the actions of the protocol down to the individual block level. A powerful modeling tool that we have created has aided us in this process. Our tool, a High Level Timed Petri Net Simulator, can be used to run the HLTPN model, thus confirming the basic correctness of the cache coherence protocol. In addition, with long simulation runs, estimates of the performance of the proposed protocol can be made. The tool includes an editor for the creation of large Petri net models, and will be described in a paper to be sent for publication in the near future.



Global Bus
Cluster 1
Cluster 2
OOO
Cluster N

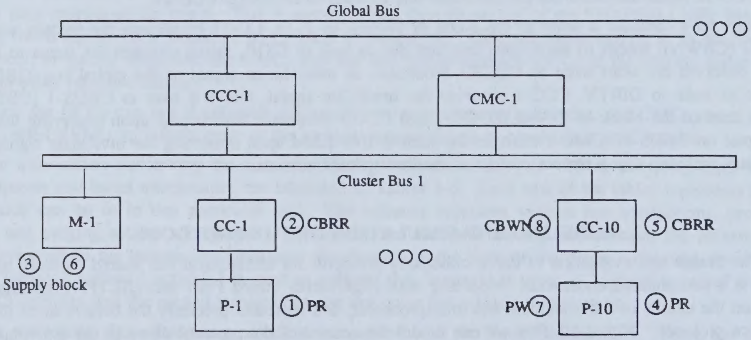
a) Clusters of processor on Global Bus.



Global Bus
OOO
OOO
CCC-n
CMC-n
Cluster Bus 1
M-n
CC-1
CC-M
OOO
P-1
P-M

b) Single cluster.

Figure 1. The hierarchical bus based multiprocessor of N clusters and M processors per cluster.



Global Bus
OOO
CCC-1
CMC-1
Cluster Bus 1
M-1
③ ⑥
Supply block
CC-1
② CBRR
CBWN⑧
CC-10
⑤ CBRR
OOO
P-1
① PR
PW⑦
P-10
④ PR

Figure 2. Typical scenario of intra-cluster sharing. Processor P-1 reads block, processor P-10 reads block then writes into block, passing new data to P-1.

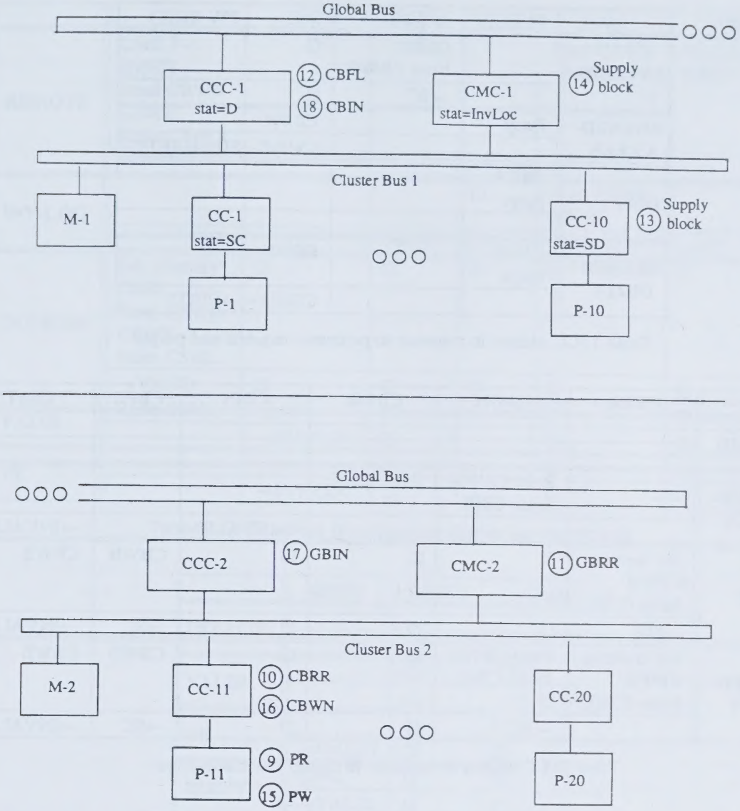

Global Bus
CCC-1
stat=D
⑫ CBFL
⑱ CBIN
CMC-1
stat=InvLoc
⑭ Supply block
Cluster Bus 1
M-1
CC-1
stat=SC
P-1
CC-10
stat=SD
⑬ Supply block
P-10
Global Bus
CCC-2
⑰ GBIN
CMC-2
⑪ GBRR
Cluster Bus 2
M-2
CC-11
⑩ CBRR
⑯ CBWN
P-11
⑨ PR
⑮ PW
CC-20
P-20

Figure 3. Continuation of scenario in Figure 2. Remote processor P-11 secures block for read, then writes into block, invalidating block on Cluster 1. Initial status of block shown in Cluster 1 devices.

	PURGE	PR	PW
INVALID		*CBRR* Read CBWB	□
		→SC	□
SHARED-CLEAN	Purge		*CBWN*
			CSHL?→SD:→DIRTY.
DIRTY	*CBWB* Purge		
SHARED-DIRTY	*CBWB* Purge		*CBWN*
			CSHL?--:→DIRTY.

Table 1: CC actions in response to processor requests and purges

	CBRR	CBWN	CBWB	CBIN	CBFL	CBFI
INVALID						
SHARED-CLEAN		Read CBWN Raise CSHL	Raise CSHL			
				→INVALID		→INVALID
DIRTY	Inh. memory CBWB Raise CSHL	□	□	□	CBWB	CBWB
	→SD	□	□	□	→SC	→INVALID
SHARED-DIRTY	Inh. memory CBWB Raise CSHL	Read CBWN Raise CSHL	□	□	CBWB	CBWB
		→SC	□	□	→SC	→INVALID

Table 2: CC actions in response to cluster bus transactions

	GBRR	GBWB	GBIN
REMOTE			
INV-LOC		□	□
		□	□
INV-REM	Read GBWB *CBWB*	Read GBWB *CBWB*	□
	→VALID	→VALID	□
VALID	*CBRR* Raise REML Read CBWB GBWB	□	
		□	→INV-REM
VE	*CBRR* Raise REML Read CBWB GBWB	□	□
	→VALID	□	□

Table 3: CMC Actions in Response to Global Bus Transactions

	CBRR	CBWN	CBWB	CBIN	CBFL	CBFI
REMOTE	CSHL?--: (*GBRR* Read GBWB CBWB Raise CSHL).		*GBWB*		Read CBWB *GBWB*	Read CBWB *GBWB*
INV-LOC				☐	Read CBWB *GBWB*	
			→VE	☐	→VALID	→VE
INV-REM	Inh. memory *GBRR* Read GBWB CBWB Raise CSHL.	☐	☐	☐	☐	☐
	→VALID	☐	☐	☐	☐	☐
VALID	Raise CSHL		☐		☐	☐
		→INV-LOC	☐		☐	☐
VE			☐		☐	☐
		→INV-LOC	☐		☐	☐

Table 4: CMC actions in response to cluster bus transactions

	GBRR	GBWB	GBIN
INVALID			
VALID		☐	*CBIN*
		☐	→INVALID
VE		☐	☐
	→VALID	☐	☐
DIRTY	*CBFL*	☐	☐
	→VALID	☐	☐

Table 5:CCC actions in response to global bus transactions

	CBRR	CBWN	CBWB	PURGE
INVALID		☐		Purge
	REML?--: (CSHL?→VALID:→VE).	☐		
VALID		*GBIN*	☐	*CBIN* Purge
		→DIRTY	☐	
VE			☐	*CBIN* Purge
	REML?→VALID:--.	→DIRTY	☐	
DIRTY			CSHL?--:Purge.	*CBFI* Purge
			CSHL?→VALID:--.	

Table 6: CCC actions in response to cluster bus transactions

References:

[1] J. Archibald and J. L. Baer, "Cache coherence protocols: Evaluation using a multiprocessor simulation model," *ACM Transactions on Computer Systems*, Vol. 4, No. 4, pp. 273-298, Nov. 1986.

[2] Futurebus Working Group of the Microprocessor Standards Committee, "P896.2/D1.0 Futurebus P896.2 Specification", Draft 1.0, April 15, 1988.

[3] A. W. Wilson, "Hierarchical cache/bus architecture for shared memory multiprocessors," in *Proceedings of the* 14^{th} *International Symposium on Computer Architecture*, pp. 244-252, June 1987.

[4] J. L. Baer and W. H. Wang, "On the inclusion properties for multi-level cache hierarchies," in *Proceedings of the* 15^{th} *International Symposium on Computer Architecture*, pp. 73-80, 1988.

[5] J. K. Archibald, "A cache coherence approach for large multiprocessor systems," in *Proceedings of the 1988 International Conference on Supercomputing*, pp. 337-345, July 4-8, 1988.

[6] T. L. Sterling, A. J. Musciano, E. Y. Chan, and D. A. Thormae, "Effective implementation of a parallel language on a multiprocessor," *IEEE Micro*, Vol. 7, No. 6, pp. 46-62, Dec. 1987.

[7] C. Scheurich and M. Dubois, "Correct memory operation of cache-based multiprocessors," in *Proceedings of the* 14^{th} *International Symposium on Computer Architecture*, pp. 234-243, June 1987.

[8] E. M. McCreight, "The Dragon computer system: An early overview," in *NATO Advanced Study on Microarchitecture of VLSI computers*, pp. 83-101, July 1984.

[9] P. Sweazey and A. J. Smith, "A class of compatible cache consistency protocols and their support by the IEEE Futurebus," in *Proceedings of the* 13^{th} *International Symposium on Computer Architecture*, pp. 414-423, June 1986.

A PERFORMANCE STUDY FOR PARALLEL COMPUTATION ON LOOSELY-COUPLED MULTIPROCESSORS

Lishing Liu
Jih-Kwon Peir

Computer Science Department
IBM T.J. Watson Research Center
Yorktown Heights, NY 10598

Douglas Logan

IBM DSD
Kingston, NY 12401

Arthur Bross
David Shen

IBM GTD
East Fishkill, NY 12524

Abstract

This paper describes a methodology of performance evaluation on parallel systems. Performance analysis in this area is often difficult without detail simulation. The methodology allows us to define a workload with certain flexibility and to assess performance of different multiprocessor designs.

Introduction

In the past decade a large number of multiprocessor systems have been proposed to support parallel numeric computations. There are two major approaches in constructing such systems: *tightly* and *loosely* coupled several processors together. Currently, most of the commercially successful machines, such as Cray X-MP, IBM 3090 [1,2], use the tightly-coupled approach. In such systems, a number of processors directly share the main memory under the same operating system. The machine performance relies heavily on the bandwidth of the centralized main memory, and hence, the number of processors in this approach is usually small. The other approach, with considerable less hardware complexity, is to loosely coupled multiple processors through communication channels. A loosely Coupled Array of Processors (lCAP) developed at Kingston [3,4] and some other hypercube machines [5] belong to this category. In contrast with tightly-coupled systems, data sharing (communication) between processors may have to pay certain software set-up overheads. Therefore, this type of systems is more suitable for coarse-grain parallelism.

In this study, we focus on the understanding of the loosely-coupled approach for future supercomputing. The primary goal of our research is to provide a methodology for evaluating performance and assessing different trade-offs in designing such coupled multiprocessor systems.

From application software point of view, the speedup of a job running on a parallel system with n processing elements can be expressed as a function of the fraction of computation spent in two modes of computing: sequential and parallel; $1 \div ((1-f)+\frac{f}{n})$, where f is the fraction of parallelized code in a program. This is the well-known Amdahl's Law. The sequential portion $(1-f)$ is a constant independent of the number of processors n. Thus, in order to achieve high efficiency, parallel computation with a small number of high-performance processors is widely adopted. However, Amdahl's law is overly optimistic in that it does not consider the overhead incurred during parallel computation. This overhead comes primarily from data communication, synchronization, and load balancing among processors. To fully understand the performance, We emphasis on estimating this overhead on various loosely-coupled environment.

Methodology

At the beginning of our study, we developed a methodology which allows us to define a workload and to do performance analysis. Trace-driven simulation used for studying most commercial system designs may not be suitable here because of the long computation in most scientific programs. However, studies show that many numeric applications have regular computational structure. We propose a methodology which simplifies the performance analysis for certain computations with high degree of flexibility. The methodology includes three steps.

- Benchmarking and Workload Characterization
- Performance Modeling and Validation
- Performance Projection.

We will illustrate each of the steps in the following sections with a case study.

Benchmarking and Workload Characterization

The first step involves benchmark selection, parallelization, and characterization. In order to understand the communication and sharing overhead associated with parallel computation on a coupled multiprocessor system, it is important to select a benchmark that requires a demanding test of data communication among the processors. A fluid dynamics code (VA3D) from NASA was chosen for this purpose. VA3D is a three-dimensional, implicit, approximate factored algorithm [6,7] solving the EULER equations or the Navier-Stokes equations with a thin layer approximation. The approximate factored algorithm is a particular form of the Alternating Direction Implicit (ADI) algorithm after Douglas and Gunn [8]

This benchmark is of interest to us for two reasons. First, the decoupling of spatial coordinates provides the source of parallel computation. Second, the sweeping through the data base from different directions requires extensive data communication among processors. This is because, when executing the ADI algorithm, each processor computes (both read and write) one region of data in one dimension. Then, it computes on another region of data in the next different dimension. This extensive degree of data sharing makes us believe that if a particular parallel system can perform well on this workload, it will probably perform well on most other workloads.

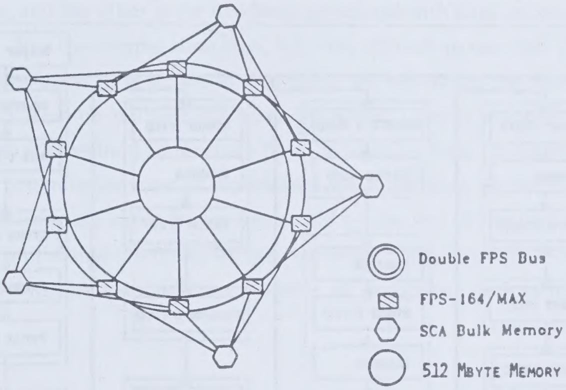


Figure 1. The organization of ICAP system

In this step, we need a parallel system to do benchmarking. We chose ICAP because of the availability of hardware and the proven Fortran precompiler for parallel computation. LCAP consists of ten Floating Point System(FPS) processors with an IBM system as a host. Communication and synchronization between processors can go through a set of locally shared memory (made by SCA company), a global 512 Mbyte shared memory, or a double-ring bus (Figure 1). The bandwidth of both shared memories is 44 Mbytes per second. In our study, we used the global shared memory as the communication media among processors. Currently in IBM Kingston, there are two such systems: ICAP-1 and ICAP-2. The primary difference between these two systems is the attached processors: FPS-164s for ICAP-1 and FPS-264s for ICAP-2. The peak performance of those two processors are 11 and 38 Mflops respectively.

Next, we need to parallelize VA3D for the ICAP Fortran precompiler. A fundamental issue in parallelizing sequential programs is to find portions of code which can be partitioned into independent execution pieces. In parallelizing VA3D, we were able to find 97% of parallel code in terms of execution time (all of them are parallel loops) without any substantial effort in restructuring the original program. The parallel model of computation on ICAP can be viewed as follows. FORK and JOIN type construct is provided in the IBM host which is responsible for job initialization and termination. During parallel execution among FPS processors, each processor will do a region of computation, then it exchanges data with other processors. Synchronizations are required to guarantee proper data dependence. This data communication and synchronization is accomplished through the global shared memory.

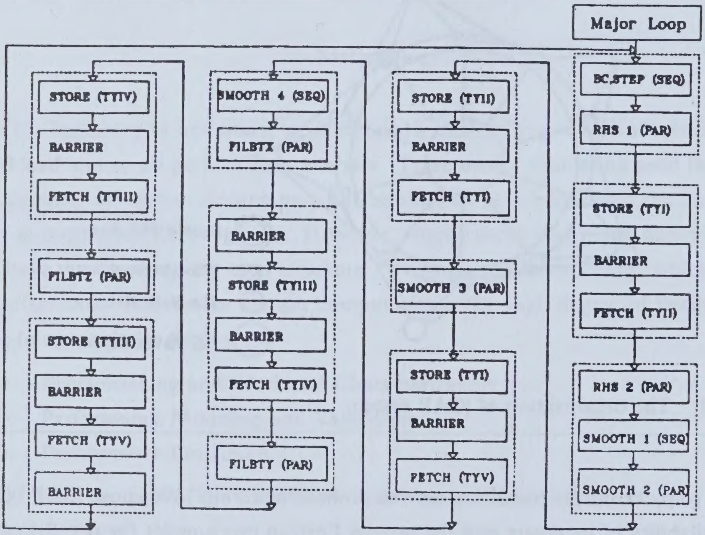

Major Loop
STORE (TYIV)
BARRIER
FETCH (TYIII)
FILBTZ (PAR)
STORE (TYIII)
BARRIER
FETCH (TYV)
BARRIER
SMOOTH 4 (SEQ)
FILBTX (PAR)
BARRIER
STORE (TYIII)
BARRIER
FETCH (TYIV)
FILBTY (PAR)
STORE (TYII)
BARRIER
FETCH (TYI)
SMOOTH 3 (PAR)
STORE (TYI)
BARRIER
FETCH (TYV)
BC,STEP (SEQ)
RHS 1 (PAR)
STORE (TYI)
BARRIER
FETCH (TYII)
RHS 2 (PAR)
SMOOTH 1 (SEQ
SMOOTH 2 (PAR

Figure 2. The computational flowchart of VA3D

Figure 2 shows the detail computation flow of the parallelized VA3D. This major loop iterates 200 times. In each iteration, the approximate factorization results in three one-dimensional operators, each of which contains derivatives in only one spatial direction. Central differencing yields block tri-diagonal matrices which are inverted in each spatial direction (FILBT). The boundary conditions (BC), right-hand side operator (RHS), and smoothing (SMOOTH) are performed explicitly. There are six computational regions represented by their major subroutine names. PAR or SEQ indicates if the routine is performed in parallel or replicated sequentially by all processors. There are also six data communication regions in which STORE and FETCH represent the data movements to and from the shared memory. Six BARRIER synchronization before FETCH are used to guarantee the write-before-read synchronization. On the other hand, only two BARRIERs before STORE are required for the read-before-write synchronization.

Performance Modelling

The second step is to construct the performance model. The total execution time of the parallelized VA3D can be divided into two parts. One is the time required for the real computation, and the other is the overhead associated with data communication and synchronization. For the computation part, it is very difficult to calculate the exact time without detail simulation. However, in this study we emphasize the sharing behavior of parallel computation on multiple systems. Therefore, to serve our purpose, we used a simple extrapolation method to estimate the computation time. From benchmarking, we obtained the computation time of the selected VA3D running on the ICAP-1 system. The computation time can easily be parameterized by the size of the problem (i.e. loop bounds). Based on the measurement, we can extrapolate larger size problems computed on new coupled systems with different number of processors and various processor speeds.

The major task is to model the shared memory access for data communication and synchronization. There have been various studies on memory design and modeling over the years [9,10,11,12]. For our purpose, we developed a careful analytical solution based on the following observations. Each physical data movement between the shared memory and processor consists of three stages:

- ***Setup Stage*** - Certain software or microcode interfaces perform necessary initialization for the I/O request. We assume a constant setup time t_S for each access protocol.
- ***Queuing Stage*** - After setup, the request is queued till the memory is ready to service the actual data transfer. The time penalty for the queuing depends upon the contention situation at the moment. Primarily the queuing is to wait for an available memory port.

- ***Transfer Stage*** - When the memory is ready to service the transfer of B_i, successive units are transferred in successive cycles. Transferring a block B_i with b units will be charged with $b \times t_T$ time.

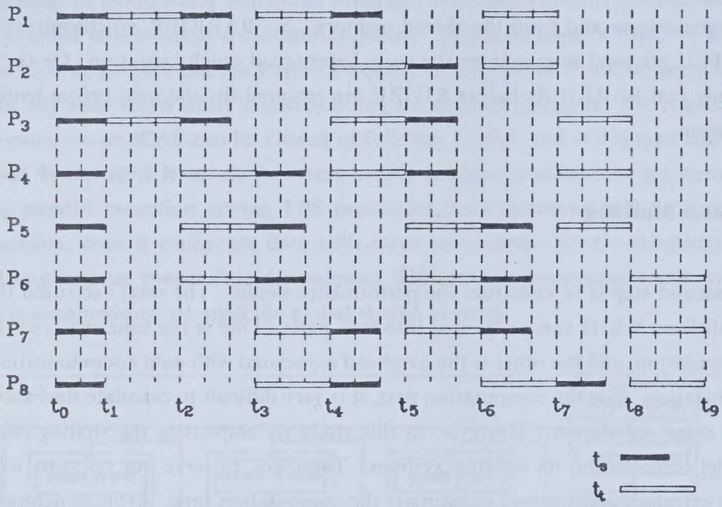


Figure 3. A shared memory access example

For example, in Figure 3 there are 8 processors each of which issues 3 requests to a shared memory with degree of concurrency 3. Each request consists of a setup and a transfer stages. A request may enter a queueing stage if the shared memory is busy servicing other 3 processors. The total elapsed time (t_9) can be computed as follows.

$$t_S + \left\lceil \frac{3 \times 8}{3} \right\rceil \times t_T$$

Based on similar method, we developed formulas to compute the elapsed time for different cases of shared memory access.

The model was then validated against measurements for the ICAP system as shown in Figure 4, where P indicates the time projected by our model. The results indicated that our model overestimated the total elapsed time by about 9%. However, for the sharing overhead, our model model was very accurate (more than 99%). The overestimation comes from the computation part that is extrapolated from the reduced size problem.

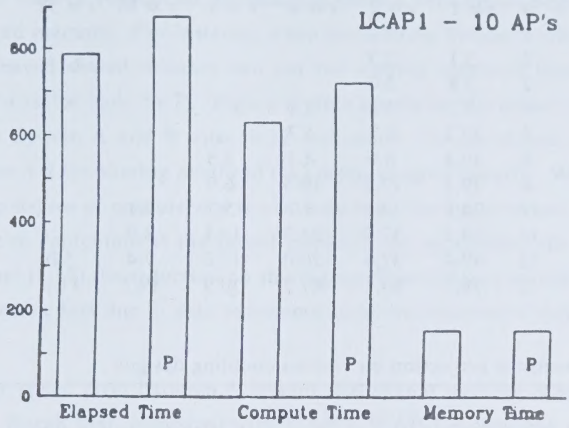


Figure 4. Model validation against ICAP-1 measurement

In the last step, we used the model to predict the performance of the selected workload in various coupling environments. We also assessed different design trade-offs by varying different parameters. We will discuss the projection in the next section.

Performance Analysis

We did performance projections on two coupling systems: A and B. The speeds of the processor and the shared memory in System A and B are the same as the speeds of the processor and shared memory in ICAP-1 and ICAP-2 respectively. Figure 5 shows the results of the two systems with various coupling designs, where sharing overhead is represented by the percentage of the total execution time. We can make several key observations based on the following parameters: (1) Number of processors (n), (2) Degree of concurrency at the global shared memory (r) (It means that at any given time, the memory can serve up to r requests.), (3) Relative speed ratios of processor and shared memory. Notice that in Figure 5, we projected the performance of two systems with the same speeds of set-up (t_S) and transfer (t_T) as in the ICAP-1 and the ICAP-2 systems. We can easily vary these parameters to predict the performance of other system configurations.

	n	$r = 1$	$r = 2$	$r = 4$	$r = 8$	$r = 16$	$r = 32$
A	2	3.1	2.9				
B	2	5.8	5.0				
A	4	5.3	3.8	3.1			
B	4	13.2	8.2	5.7			
A	8	10.8	6.6	4.5	3.5		
B	8	29.2	17.2	10.5	6.9		
A	16	24.8	14.7	8.7	5.7	4.1	
B	16	54.7	37.8	23.3	14.1	8.9	
A	32	49.4	32.8	20.0	12.0	7.4	5.0
B	32	78.1	64.1	47.2	30.9	19.2	11.8

Figure 5. Performance projection on various coupling designs

(1) It is clear that, as number of processor grows, the sharing between processors is causing higher overhead for two reasons. First, the sharing load goes up with the number of processors. (In this workload, for example, there are cases that each processor requires to fetch entire array from the shared memory.) Second, there are more contention in accessing the shared memory for larger number of processors. The figure shows that when the number of processors doubled in both systems the sharing overheads are also roughly doubled.

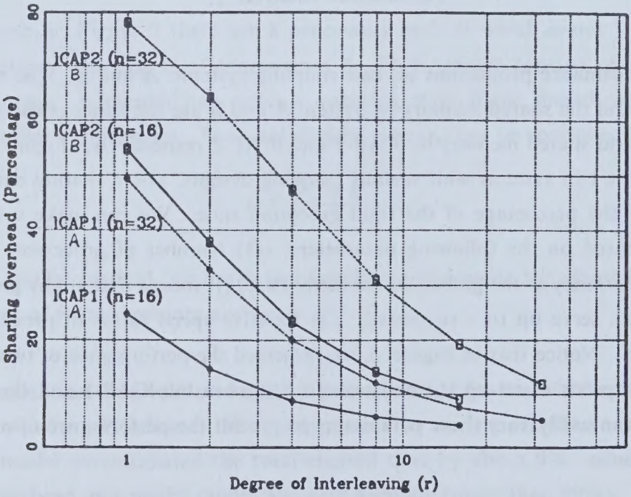


Figure 6. Sharing overhead vs. degree of interleaving

(2) We observe substantial performance benefit with higher degree of concurrency of the shared memory. For instance, when considering System B with 32 processors, a fully interleaved shared memory can cut the sharing overhead from 78.1% down to 11.8% (by a factor close to 7). Figure 6 plots results on the impact of degree of interleaving for System A and B with 16-32 processors. As the degree of interleaving increases from 1-4 the sharing overhead (%) drops roughly linearly. Which demonstrates that, as the degree of concurrency is relatively low, sharing overhead in such systems is mainly due to contention at the shared memory. As we increase the degree of concurrency further (8-32) the reduction on sharing overhead flattens up slightly, in which case the performance loss due to data movement setup becomes more visible.

(3) The speed ratio between processor and shared memory influences the sharing overhead. Recall that, compared with ICAP-1, ICAP-2 system has more than 3 times faster processors while the shared bulk memory is about .9 times slower. The sharing overhead in System B should be identical to that in System A if the shared memory performance were scaled up according to the processor speed. It is seen in Figure 5 that the sharing overhead is higher by a factor between 60-200%. The reason that we do not observe 3 times higher sharing overhead in System B, compared with System A, is that the data movement setups are also roughly 3 time faster in System B.

Summary

We proposed a methodology for performance study of parallel computation on loosely-coupled multiprocessor with a shared storage. This method allowed us to evaluate the performance of different coupling strategies and design trade-offs, which would have otherwise been very difficult to assess. From the performance prediction, we can conclude that under certain assumptions and design enhancement, the sharing overhead of the carefully selected workload on a loosely coupled environment can be greatly reduced (to less than 5% in the ICAP-1 environment). Because the requirement for hardware interconnection is usually simpler than the tightly-coupled approach in certain environment, the loosely-coupled approach provides us a possible alternative for future supercomputing.

Acknowledgement

The authors wish to thank Enrico Clementi, John Detrich, Dave Folsom for their support of the ICAP facility. We also thank Frederique Giraud for her helpful discussions.

References

1. O. Lubeck, J.Moore, and R. Mendez, "A Benchmark Comparison of Three Supercomputers: Fujitsu VP-200, Hitachi S810/20, and Cray X-MP/2," *IEEE Computer*, Vol. 18, No. 12. Dec. 1985, pp. 10-29.
2. S. Tucker, "The IBM 3090 System: An Overview," *IBM System Journal*, Vol. 25, No. 1, 1986, pp. 4-19.
3. E. Clementi et al, "Large Scale Computation on a Scalar, Vector and Parallel Supercomputer," IBM Kington TR KGN-77, Aug. 1986.
4. S. Chin et al, "Parallel Computation on The Loosely Couple Array of Processors: A Guide to The Precompiler," IBM Kingston TR KGN-42, Nov. 1985
5. G. C. Fox, S. W. Otto, and A. J. G. Hey, "Matrix Algorithms on a Hypercube I: Matrix Multiplication," *Parallel Computing*, Vol. 4, No. 1, Feb. 1987, pp. 17-32.
6. T. H. Pulliam, "Air3D," NASA/Ames.
7. J. L. Martin, "Performance Evaluation of Supercomputers and Their Applications," *Proc. of IBM Symp. on Parallel Processing*, Aug. 1986.
8. Douglas, and Gunn "A General Formula of Alternating Direction Method," *Numerical Methods*, Vol.6, No.15, 1964.
9. D. H. Lawrie, and C. R. Vora "The Prime Memory Systems for Array Access," *IEEE Trans. on Computers*, Vol. C-31, No. 5, May 1982, pp. 435-442.
10. H. Garcia-Molina, R. J. Lipton, and J. Valdes "A Massive Memory Machine," *IEEE Trans. on Computers*, Vol. C-33, No. 5, May 1984, pp. 391-399.
11. A. Norton, and G. F. Pfister, "A Methodology for Predicting Multiprocessor Performance" *Proc. 1985 Int'l Conf. on Parallel Processing*, Aug. 1985, pp. 772-781.
12. T. Cheung, and J. E. Smith, "A Simulation Study of the CRAY X-MP Memory System," *IEEE Trans. on Computers*, Vol. C-35, No. 7, July 1986, pp. 613-622.

A MAPPING ALGORITHM FOR MULTIPROCESSOR ARCHITECTURES

S. YALAMANCHILI and D.T. LEE
Honeywell Systems and Research Center
3660 Technology Drive, MN65-2100
Minneapolis, Minnesota 55418

ABSTRACT

In this paper we present an algorithm for mapping a set of parallel tasks onto a homogeneous multiprocessor architecture. The algorithm is based on a combinatorial optimization procedure known as simulated annealing. A simulation of the mapping algorithm has been used to evaluate the quality of mappings achieved with several multiprocessor architectures. These include various forms of two-dimensional arrays and hypercube architectures. We discuss the present form of the algorithm, the results of our simulation studies, and future extensions.

INTRODUCTION

One of the most important problems in the design and use of multiprocessor architectures is resource management. In addition to the management of conventional resources such as memory and processor time, it is also necessary to manage resources at the level of processing elements (PEs), and communication links. Specifically, two aspects of this problem must be addressed. The first is the efficient allocation/deallocation of PEs. We envisage an environment where multiple jobs will be executing on a single multiprocessor machine and where each job comprises concurrently executable (within precedence constraints) tasks. These jobs would arrive at indeterminate intervals with varying computational requirements, and each job may not use, or require, the complete machine. Therefore, it is important to be able to allocate partitions of the machine to each job. The second problem in resource management is the assignment of tasks to the allocated PEs. This is the well-known mapping problem [1]. If naive assignments of individual tasks to PEs are made, the overhead of intertask communication can significantly degrade performance, even to the point of nullifying the gains of speed-up.

In this paper, we outline a general approach to solutions of the mapping problem, which is applicable to a wide range of multiprocessor systems. The next section presents an overview of our technical approach and is followed by a description of a solution to the mapping problem. We discuss the simulated annealing approach, the details of the mapping algorithm, the simulation experiments, and the results obtained. The results include simulation experiments with both hypercube and two-dimensional array structured multiprocessors. We conclude with a brief description of directions for future research.

TECHNICAL APPROACH

Our technical approach is predicated on the overall view that the relationships between the system software components are as depicted in Figure 1. The solution to a specific problem is translated into a parallel program. The program is compiled to produce the executable code and a problem graph. The problem graph is a weighted/directed graph, $G_p = (V_p, E_p)$, where the nodes, V_p, correspond to a set of parallel tasks, and the edges, E_p, represent the communication between tasks. The weights associated with the nodes represent processing requirements. For example, the weights may represent the number of arithmetic and logical operations to be performed or the number of instructions to be executed. Similarly, the weights on the edges represent the volume of communication, e.g., in bytes, between communicating tasks. A parallel architecture is represented by a weighted/directed graph, $G_a = (V_a, E_a)$. The set of nodes, V_a, corresponds to the PEs. Each PE usually consists of a processor, a local memory, and a communication interface. The node weights indicate the speed of the PEs at the nodes, and the edge weights indicate the communication bandwidth of the respective links. This formulation is intended for distributed memory architectures.

As shown in Figure 1, the Scheduler is responsible for determining the job that will have next access to the multiprocessor system. It is assumed that modifications to conventional scheduling disciplines are applied to schedule these jobs. Our concern is with the remaining three modules–the Resource Manager, the Mapper, and the Projector. The Resource Manager and the Mapper have a producer-consumer relationship. The Resource Manager allocates PEs, i.e., partitions of the multiprocessor, to incoming jobs. The Mapper assigns the parallel tasks to the PEs allocated by the Resource Manager. The Mapper can also allocate tasks to PEs in a virtual multiprocessor architecture. For example, consider a hypercube architecture. This virtual hypercube can be much larger than the allocated subcube in the same way that virtual address spaces exceed physical memory. In this case, the Projector takes the assignment of tasks to PEs in the virtual hypercube and to the set of allocated PEs and projects the larger virtual cube on the smaller, available cube.

Only the Mapper module is described here. We described the algorithms for the other modules in an earlier report [2].

THE MAPPING PROBLEM

Finding the optimal assignment for a system with four or more processors is NP-complete. In its most general form, the mapping problem is computationally equivalent to the graph isomorphism problem, one of the classical unsolved combinatorial problems. Finding an exact, efficient (i.e., in polynomial time) algorithm for solving this problem for arbitrary graphs appears unlikely. The objective is to develop a set of heuristics that can be used to map a set of parallel tasks onto a parallel architecture in order to maximize system utilization and throughput.

Bokhari proposed a mapping strategy based on the cardinality of the mapping [1]. When a problem graph was mapped onto a system graph, the number of problem edges that fell on the system edges was the cardinality of the mapping. Iterative improvement and probabilistic jumps

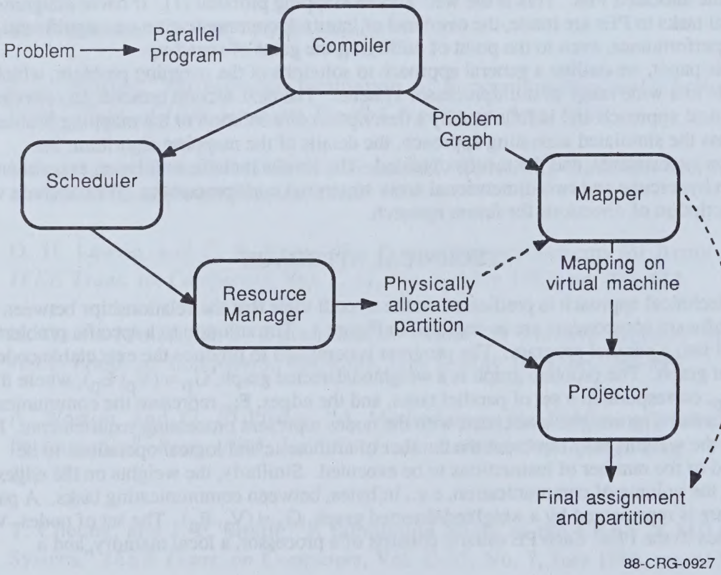


Figure 1. Functionality of the system resource allocation process

were used to search for the best mapping, i.e., the one with the largest cardinality. There are, however, several limitations to this strategy. Routing between the tasks mapped to nonadjacent PEs is not allowed, and is, indeed, ignored. The strategy focuses on communication, while ignoring computation issues such as load balancing. Also lacking is the ability to accommodate weights on the edges and nodes. Finally, Bokhari's strategy does not explicitly consider the relative speed of the PEs or of the communication links. Lee and Aggarwal's recent work focuses on some of these limitations, providing an algorithm that is a generalization of the one proposed by Bokhari [3]. However, this algorithm requires a complete static scheduling of all interprocessor communication. Instead of some orderly progression toward a global minima, local minima are avoided by choosing multiple starting points.

The mapping algorithm described in the next section allows for a combination of PE speeds and communication bandwidths, places no restrictions on the relative number of tasks or PEs, and is based on an approach that, under certain conditions, can be guaranteed to find a global optimum. Furthermore, communicating tasks are not constrained to being mapped onto adjacent PEs, and communication issues are considered without requiring a complete static scheduling of interprocessor communication. Many variants of this algorithm can be created by altering the conditions controlling the search for the "best" solution. The next section describes one variant that we have found successful.

Simulated Annealing

We propose to achieve the task assignment with a combinatorial optimization procedure known as simulated annealing. The simulated annealing approach is a probabilistic, hill-climbing, Monte Carlo optimization technique based on Metropolis' algorithm [4]. It simulates the slow cooling of a many-body system (such as crystals and spun glass) from a random arrangement at high temperature (melted crystals) to an orderly minimum energy state at absolute zero. Simulated annealing has been demonstrated to give good heuristic solutions for NP-complete problems such as the traveling salesman problem and other combinatorial optimization problems. It generally outperforms simple iterative improvement algorithms and is widely used in scheduling and network problems of practical importance.

The Metropolis algorithm was developed to provide an efficient simulation of a collection of atoms in equilibrium at a given temperature using statistical mechanics. The objective was to find the ground state of the system in the limit of low temperature. In each step of this algorithm, an atom is given a small random displacement, and the resulting change in the energy of the system (ΔE) is computed. If $\Delta E \leq 0$, the displacement is accepted, and the configuration with the displaced atom is used as the starting point for the next step. If $\Delta E > 0$, the displacement is accepted probabilistically according to $P(\Delta E) = \exp(-\Delta E/kT)$, where k is the Boltzmann's constant and T is the temperature. By repeating these steps many times, this algorithm simulates the thermal motion of atoms in thermal contact with a heat bath at temperature T.

The simulated annealing approach makes use of the above technique to heuristically find the global minimum of a multivariable function. By defining configurations as a set of variables or parameters and using the appropriate energy function, the Metropolis algorithm can be used to generate a population of configurations for a given optimization problem at some effective temperature. This temperature is simply a control parameter expressed in the same unit as the energy function.

The simulated annealing process starts out by "melting" the system being optimized at a high effective temperature. The temperature is then slowly lowered by stages until the system "freezes" and no further changes occur. At each temperature, the simulation must proceed long enough for the system to reach a steady (equilibrium) state. The annealing schedule is the sequence of temperatures and number of configurations accepted in reaching equilibrium at each temperature.

Unlike greedy algorithms and other simple iterative improvement techniques, the simulated annealing approach provides a mechanism for the search procedure to get out of local minima. It is also generic and universal in nature, not problem-dependent like other heuristic techniques, and is thus simple to implement.

The following is a short outline of the algorithm:

1. Define a multivariate energy function $E(x) = E(x_1, x_2, \ldots, x_n)$ in which $x_1, x_2, \ldots, x_n$ represent the state of the combinatorial system.
2. Start with a random state at an appropriate high temperature T_H.
3. Reduce the previous temperature by a small factor, i.e., $T_i = \alpha T_{i-1}$.
4. At each temperature T_i, repeat the following moves until a set of equilibrium conditions are met:
   - Select a random change of state from the current state x. Let the changed state be x'.
   - Compute the change in the energy function, $\Delta E = E(x') - E(x)$.
   - If $\Delta E \leq 0$, accept the change, i.e., x' becomes the new state.
   - If $\Delta E > 0$, accept the change with probability $e^{-\Delta E/T}$. Note that at $T = 0$, the simulated annealing approach becomes an iterative improvement technique where only $\Delta E < 0$ will result in x' being accepted.
5. Repeat steps 3 and 4 until a low enough temperature $T_L \simeq 0$ is reached and a set of terminating conditions are met.

Different algorithms can be realized based on the choice of terminating conditions, equilibrium conditions, initial temperature, and temperature decay function. Although simulated annealing is motivated by an analogy to a physical process and explained using statistical mechanics, this approach can also be described in terms of Markov chains. The implementation of simulated annealing is currently an active area of research. Several modifications to the basic approach have been proposed in the literature for improved performance.

We propose an energy function that allows specification of each of the above parameters and that is applicable to different combinations of problem graph and system graph sizes. This energy function consists of two parts, computation energy and communication energy. The objective is to minimize the sum, since ideally, each component would be as small as possible.

The computation energy, E_1, can be expressed as

$$E_1 = \sqrt{\frac{\sum_{i=1}^{N} (l_i * p_i)^2}{N'}} \tag{1}$$

where

l_i = processing load of processor i
p_i = processing speed of processor i, e.g., MIPS
N = number of processors in the system
N' = number of processors with nonzero processing load for a given mapping

For a fixed system processing load, E_1 is minimized when perfect load balancing is achieved among the PEs. Load-balancing means the minimization of PE idle time. By proper distribution of tasks and routing of the associated communication links, the PEs can be used in an efficient manner. The presence of the p_i variables allows heterogeneous computing elements to be taken into account in the system.

The communication energy, E_2, can be expressed as

$$E_2 = \frac{\sum_{i,j=1}^{M} e_{ij} * S_{ij} * \omega_{ij}}{N''} \tag{2}$$

where

e_{ij} = the weight of the edge link between nodes i and j of the problem graph, i.e., the required communication bandwidth from problem node i to problem node j
S_{ij} = the number of system links traversed in order to transfer data from problem node i to problem node j
ω_{ij} = the bandwidth of the system links connecting problem node i to problem node j
M = number of edges in the problem graph
N" = number of links (edges) in the system graph that can be used by the problem graph

The communication energy, E_2, represents the average amount of communication traffic that is imposed on the system links by the problem graph. Locality of communication is encouraged. Note that we do not need to specify the exact communication path for e_{ij}. We only have to determine the number of physical links required for e_{ij}, which is much easier to compute and does not require complete static scheduling of interprocessor communication as do other systems [3]. And if the problem edges can be mapped directly, one-to-one, onto the system edges, this term is minimized. The term N", the total number of available physical links in the system, is used as a normalization constant. For a hypercube system with N nodes, N" = N $\log_2 N$. This normalization is necessary, for example, in cases where a set of tasks may only use a subcube of eight nodes in an allocated subcube of 16 nodes.

The energy function to be used in evaluating the mapping can now be represented as

$$E = \nu E_1 + (1 - \nu) E_2 \tag{3}$$

where ν is a user-specified weighting factor for adjusting the relative importance of computation versus communication in the performance of a system.

As proposed here, the energy function, E, is applicable to the general mapping problem. There are no assumptions or limitations on the sizes, configurations of the problem graph, or the system graph. It is applicable to cases where the number of tasks is greater than, equal to, or less than the number of the PEs. The presence of the parameters p_i (processor speed) and ω_{ij} (bandwidth) enables the inclusion of heterogeneous PEs and communication links in the system.

Simulation Experiments

The general procedure for simulated annealing has been described in the previous section. We define the implementation-dependent aspects here.

State Change–Let i be the index specifying the array of tasks in the problem graph, with i = 1, . . . , m. The state x of the system is defined as the vector $(x_1, x_2, \ldots, x_m)$ where x_i is the PE onto which task i has been mapped. We employed five different strategies for moving tasks.

1. Random selection of an entire new state without correlation to the current state
2. Random displacement of each task to another PE, achieved by adding a displacement vector to the current state
3. Reassignment of each task to a randomly chosen neighboring PE (even if no movement is required)
4. Reassignment of only one task and mapping it to a randomly chosen PE
5. Reassignment of only one task to a randomly chosen neighbor

These are the different strategies we can use to move in the search space and to converge to the final solution. Moving task strategies 1 through 3 represent parallel state changes, while 4 and 5 are sequential state changes. In addition, moves 3 and 5 are referred to as nearest-neighbor state changes, while the remaining three do not exploit the interconnection topology. All of the moves maintain "detailed balance," i.e., the probability of moving from state i to state j is the same as a transition from state j to state i. After experimenting with a number of strategies, we found that two of them were most effective. We used strategy 2 to allow for a greater range of movement, and changed to strategy 5 at lower temperature for "fine tuning" of the mapping. We specified the changeover to occur after the initial temperature had been decreased by an order of magnitude.

Initial Temperature–The initial state is chosen at random, and the annealing process begins at a sufficiently high temperature. We typically set the initial temperature as twice the energy function value of the initial state.

Equilibrium State–For a temperature T, the state has reached equilibrium if the number of acceptances of change at temperature T passes a predetermined threshold (we set it at 20 * N) or if the number of iterations at temperature T exceeds 100 times the number of acceptances. The latter condition is for low temperatures, where few acceptances are expected. At low temperatures, the running time is quite high because many candidate moves are rejected before a move to a new state is accepted. In addition, the number of accepted states drops off rapidly at lower temperatures. We have incorporated this into our test for equilibrium by decaying the number of acceptances by 0.9 at successive temperatures. This number is decayed to a minimum of 60 at low temperature. *The result is that the convergence to equilibrium at low temperatures is comparable to convergence at high temperatures.* This has been achieved without any apparent compromise in the quality of the results. This is important, since most implementations of simulated annealing have suffered from very slow convergence rates at low temperatures. When equilibrium is reached, a new temperature T' is computed, and the process of generating new states is repeated. We have set the temperature decay factor, α, as 0.9. *In all of our experiments, the number of iterations (and temperatures) required to reach the final state is roughly the same for each experiment* (± 10%).

Termination Condition–The simulated annealing process is considered complete and the system frozen when the value of the energy function, E, has not changed for five consecutive temperatures.

Results

We have developed a simulation of the Mapper. The multiprocessor systems that are currently supported include hypercubes, four-neighbor meshes, eight-neighbor meshes, and four-neighbor wraparound meshes (torus). It became evident very early that a major problem with any approach to mapping tasks onto an architecture is the evaluation of the results. While the Mapper should be efficient for arbitrarily structured problems, knowing a priori the globally optimum assignment for the purpose of evaluation is tantamount to having solved the mapping problem. We addressed this problem in the manner suggested by Bokhari [1]. The problem graph was made identical to the system graph. For example, hypercubes were mapped onto hypercubes, four-neighbor meshes onto four-neighbor meshes, etc. Unlike existing algorithms, it is also possible to map larger problem graphs onto smaller system graphs or vice versa, e.g., a 64- (8-) node problem graph onto a 8- (64-) node system graph. We also examined cases such as the mapping of four-neighbor meshes onto hypercubes. In all of these instances it is possible to compute analytically the value of the energy function for the globally optimum assignment. *However, the Mapper does not exploit this known structure of the tasks. It only enables us to evaluate the performance of the assignment algorithm.*

In the case of the hypercube architectures, the performance of the algorithm is more sensitive to the system graph size than the problem graph size. For 8- and 16-PE systems, the global optimum was found approximately 90% of the time, dropping off to about 60% for 32-PE systems. The solutions for the remaining cases typically had energy functions whose value was within 10% to 20% of the optimum. Similar trends were observed for two-dimensional arrays, though the incidence of optimum solutions was about 10% to 20% less. We also examined other cases with a known theoretical optimum, such as mapping two-dimensional arrays onto hypercubes. Some of the results summarized in Figure 2, 3, 4, and 5 illustrate the energy function histograms of the various experiments.

In each case, the starting assignments were randomly chosen. In all of these simulations, the relative speed of communication to computation was maintained at 1. The weighting factor v was 0.5. Each of the Figures 2 through 5 shows the histograms of the energy function values of the solutions for 50 experiments. Most of the solutions have energy function values within 10% of the value corresponding to that of the optimum mapping. In these experiments the weights of

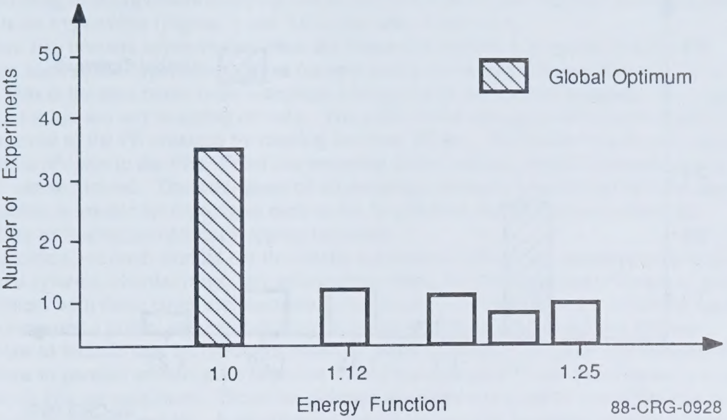


Figure 2. Mapping of 16-node hypercube onto a 16-node hypercube

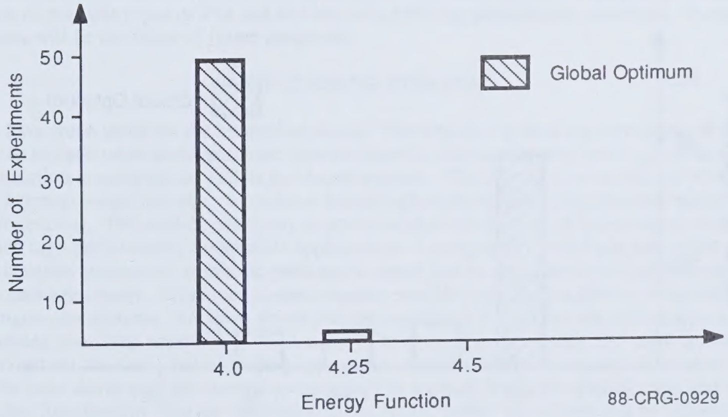


Figure 3. Mapping of 32-node hypercube onto an 8-node hypercube

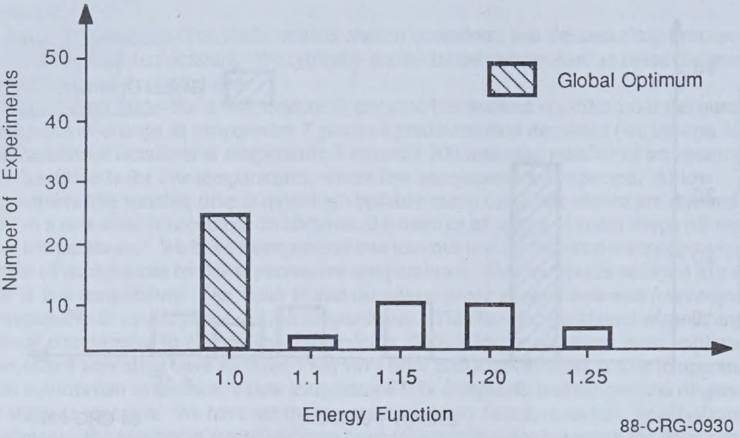


Figure 4. Mapping of 16-node mesh onto a 16-node mesh

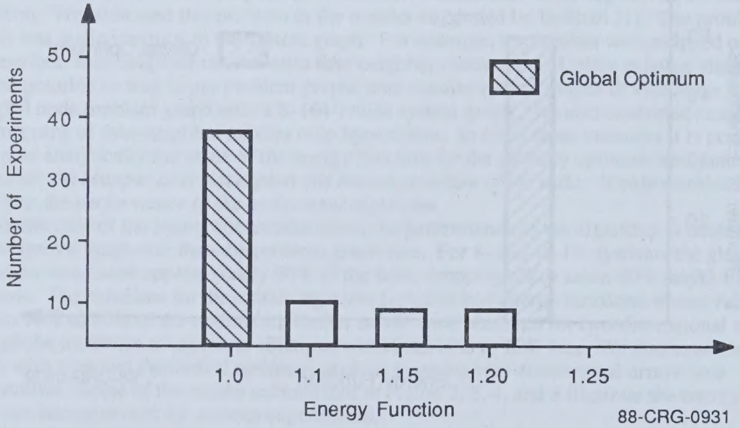


Figure 5. Mapping of 16-node torus onto a 16-node hypercube

the nodes and edges in the problem graphs are 1. The speed of the processors and the bandwidth of the communication links are also 1. Based on the energy function described earlier, the values corresponding to the optimum mapping can be computed as 4.0 for mapping 32-node hypercubes onto 8-node hypercubes (Figure 3) and 1.0 in the other three cases.

It has also become apparent that when the target architecture is a regular, symmetric topology, such as the hypercube or torus (as opposed to the mesh), this algorithm performs much better. This is because many more mappings correspond to the optimal mapping. For example, consider a torus and any mapping of tasks. The value of the energy function is the same if every task is moved to the PE obtained by rotating the torus 90 deg. This holds true for the optimal assignment of tasks to the PEs. From one mapping that is optimal, multiple mappings that are optimal can be derived. Thus, the space of all mappings contains a number of optimal mappings. This number is greater for topologies such as the hypercubes and tori, and therefore the probability of finding an optimal mapping increases.

In addition, we have worked out the details required to include architectures such as multiple bus-based systems, chordal rings, and architectures based on interconnected clusters of buses. The problem with these target architectures is constructing test problems for which the optimum can be computed a priori, making it difficult to evaluate the performance of the Mapper. Extensions to include new architectures based on static topologies are relatively straightforward. Extensions to parallel architectures based on shared buses require a new move strategy, due to the fact that all PEs are neighbors. The strategy here is to move a process to one of the PEs on which a neighboring process resides. A neighboring process is one that corresponds to an adjacent node in the problem graph. The choice of processor is random. The move strategy attempts to place communicating processes on the same PE, thus reducing the communication requirements.

Another change to the move strategy will result in the ability to accommodate heterogenous systems. Consider an element, x_i, of the state. In general, the value of x_i is changed in accordance with the move strategy, and at any time can take on some range of values. By restricting the range of values that each component x_i can take, heterogenous systems can be accommodated. Let us assume we have two types of task modules (numeric and symbolic), as well as two types of PEs. In moving a process, those corresponding to numeric computation are only allowed to move to numeric PEs, while those corresponding to symbolic computation are only allowed to move to symbolic PEs. We can introduce the notion of types to processes and PEs and restrict the random moves to maintain compatibility between types. The processor speed and bandwidth variables in the energy function become arrays, corresponding to the existence of multiple types of PEs and to links with differing performance attributes. These extensions will be the focus of future programs.

CONCLUDING REMARKS

We have three goals for the immediate future. The first is to extend the simulation of the Mapper to include other multiprocessor systems based on commonly used static, point-to-point interconnection topologies, as well as bus-based systems. The second is to extend the Mapper to include heterogeneous systems. We believe future high-performance multiprocessor systems will be heterogeneous. This will be necessary to accommodate the number of architectures designed to provide high performance for specific applications. Examples of special-purpose architectures include numeric processors, symbolic processors, signal and image processors, and general-purpose data processors. Moreover, shared memory machines can be modeled as a special case of heterogeneous systems. We have developed the extensions we believe are sufficient to extend the simulated annealing approach to the domain of heterogeneous systems. Our third and final goal is to extend the simulated annealing approach to accommodate the dynamic allocation of tasks. We have developed the extensions necessary to support the quasi-dynamic case and will include this functionality into the simulation in our future work. We believe that by achieving the above goals and integrating the various modules shown in Figure 1, we will have an invaluable tool for researching efficient resource management approaches for multiprocessor architectures.

REFERENCES

1. Bokhari, S.H., "On the Mapping Problem," *IEEE Transactions on Computers*, Vol. C-30 (3), March 1981, pp. 207-214.

2. Yalamanchili, S., and Lee, D., "Resource Management Strategies for Multiprocessor Architectures," Technical Report, Computer Sciences, Honeywell Systems and Research Center, 1987.

3. Lee, S.Y., and Aggarwal, J.K., "A Mapping Strategy for Parallel Processing," *IEEE Transactions on Computers*, Vol. C-36 (4), April 1987, pp. 433-442.

4. Metropolis, N., et al., "Equation of State Calculations by Fast Computing Machines," *Journal of Physical Chemistry*, Vol. 21 (6), June 1953, pp. 1087-1092.

SYSTEMS WITH NEARLY ZERO DISTRIBUTED SIMULATION OVERHEAD[1]

DEVENDRA KUMAR
Department of Computer Engineering, Case Western Reserve University
Cleveland, Ohio

ABSTRACT

We identify a class of systems which can be simulated by a distributed simulator with almost no overhead messages. A system in this class is a network of processes that communicate with each other solely via messages. The communication delays in this network are allowed to be arbitrary, though finite, and the communication lines are assumed to be FIFO. This class of systems includes many cyclic systems. To show that these systems can be simulated with nearly zero overhead, we present such a distributed simulation algorithm for systems in this class. By identifying this class of systems and designing a distributed simulation algorithm specifically geared for it, we have been able to avoid almost all overhead messages. By contrast, other distributed simulation algorithms are usually designed for more general systems and they usually require too many overhead messages. Our results also show that a direct application of the scheme in [4] to our systems may result in lower overhead than the usual overhead of [4] in simulating more general systems. Moreover, our results show that in simulating this specific class of systems, the overhead of schemes in [3,4] can be reduced via simple modifications to the schemes.

1. INTRODUCTION

A *distributed simulator* for a system consists of a distributed network of processes that communicate with each other solely via messages. Each process simulates a subsystem, i.e., part of the overall system being simulated. Any such process, on simulating an event occurring in its corresponding subsystem, sends event messages to the other processes that correspond to other subsystems affected by the event. At any instant during the simulation, different processes may have simulated their corresponding subsystems up to different simulation times. No global simulation clock is maintained; synchronization among processes is achieved by including, in each event message, the time of occurrence of the event along with other descriptions of the event. The motivation behind distributed simulation is to gain a speed up over traditional sequential simulation– groups of the above processes can be executed in parallel on the different processors of a distributed system, potentially resulting in a speed up over sequential simulation.

A major problem in designing a distributed simulator is that it may deadlock. Several distributed simulation algorithms have been proposed, e.g., [2-4,9,10,23,24]; they differ mainly in the way that they handle this deadlock problem. All these algorithms utilize overhead messages to avoid, or to detect and break deadlocks, or to otherwise ensure progress of simulation. The number of overhead messages is usually large, e.g., [4,25,29] point out this large overhead problem in using the distributed simulation algorithm of [3], and a corresponding severe performance degradation of the simulator.

Note that most of the distributed simulation schemes in literature, e.g. [2-4,9,10,23,24], apply to fairly general classes of systems being simulated (in terms of correctness). We expect the performance properties of any such scheme to vary substantially depending upon the

---

[1] This work was supported by Air Force Grant AFOSR 81-0205.

characteristics of the system being simulated. For almost any scheme, one can find examples of systems for which it has a large (or, alternatively, small) amount of overhead. This kind of behavior pattern is seen even in the case of sequential simulation — e.g., event driven simulation is suitable for only a certain class of systems, and not for others.

One avenue of research would be to identify special classes of systems that can be simulated with low overhead. In the context of simulating any such specific class of systems, one may study questions like: (i) how do the various general purpose schemes compare with each other in terms of performance, (ii) does a given distributed simulation scheme have special performance properties in simulating these systems, as opposed to more general systems, and (iii) can we exploit the special properties of these systems to develop new schemes specifically tailored for this class of systems, with a better performance than above (either by modifying previous schemes or by devising entirely new schemes).

Once we have understood the simulation properties of these special classes of systems, we might be able to devise appropriate simulation schemes for more complex systems, e.g., by partitioning those systems.

In an earlier work [11,13,15], we showed that any acyclic network can be simulated via distributed simulation with nearly zero overhead. In this paper, we extend this result by identifying another class of systems which, we claim, can be simulated via distributed simulation with nearly zero overhead. This class includes many cyclic systems. The characteristics of this class of systems is defined in section 2. To prove the above claim that these systems can be simulated with nearly zero overhead, we present a corresponding distributed simulation algorithm (algorithm 1) with no overhead messages, in section 3, and sketch its correctness proof in section 4. This simulator correctly simulates all the desired events of the system being simulated; however, at the end of simulation, some of the simulator processes could be waiting to receive messages, unaware of the fact that the simulation is over and no such messages will arrive in the future. In section 5, we devise our algorithm 2 by adding some overhead messages to algorithm 1, thereby enforcing normal termination of all processes in the simulator. The total number of overhead messages in algorithm 2 is very small — equal to the total number of communication lines in the system being simulated.

2. CHARACTERISTICS OF THE SYSTEMS TO BE SIMULATED

We define here the class of systems mentioned above. The characterization is at an abstract level, and many systems that do not directly satisfy these characteristics can still be modeled to satisfy them. A system to be simulated, henceforth called a *physical system*, is assumed to be a finite network of processes, called *physical processes* (or *pps* for short), that communicate with each other solely via messages. The processes in the above network are connected by one to one, unidirectional, unbounded, and FIFO (i.e., First-In-First-Out) *communication lines*. There is at most one line from pp i to pp j, denoted by the ordered pair (i,j). There is no line from a pp to itself. The communication delay between sending a message and its *arrival* at the input port of its destination pp is arbitrary (possibly zero), but finite. Messages arrived along a line may be *received* by the destination process in FIFO order at any time after their arrival. Note that the arrival of a message along a line is different from its reception by the destination process.

Each pp i executes a loop consisting of the following steps. It determines an adjacent line *l* with which it wishes to communicate. If *l* is an output line, then a message m is computed and sent out on the line *l*. If *l* is an input line, then pp i receives a message from the input port that has arrived along the line, in the FIFO order for all messages arrived along that line (if no message corresponding to the line is currently available on the input port, then the process waits until one arrives). If pp i does not wish to communicate with any line, it simply terminates. It is assumed that there is no deadlock, i.e., a permanent circular waiting, at any time in the physical system. However, note that it is possible for a pp to wait for an input message forever (e.g., if the corresponding sender process is terminated).

The objective of simulation, at an abstract level, is to record all message communications (i.e., send and receive operations) by each process, including the times at which they take place. In order to simulate the system, certain assumptions regarding the computability of events are required. These assumptions, along with several others to keep our discussion simple, are given next.

We assume that the physical system is required to be simulated for a time period [0, Z], where Z> 0 is a prespecified time. The send and receive operations of any given pp take place at distinct instants of time greater than zero (this is called the *process chronology* property). We assume that the total number of messages sent in the physical system up to (including) time Z is finite. These assumptions simplify our representation of history of messages sent or received by a pp, thereby simplifying the proof of correctness of our algorithm. Moreover, the above assumption regarding finiteness of number of messages is needed in order to complete the simulation in finite time. We define the mathematical object *message history* to be a sequence of the form $<(l_1, t_1, m_1), (l_2, t_2, m_2), \ldots, (l_r, t_r, m_r)>$ where $r \geq 0$, $l_1, l_2, \ldots, l_r$ are communication lines, $0 < t_1 < \ldots t_r$ are time values, and $m_1, m_2, \ldots, m_r$ are message values (these communication lines, time values, and message values in a message history must be different from NULL — a fictitious value, discussed later, that is distinguishable from these values in any physical system.) The t-value of a message history is defined to be the t-value of the last tuple in it, if any, or zero if it is the empty history. The history of messages sent or received by pp i up to (and including) time t in the physical system is represented by the above message history, where $0 < t_1 < t_2 < \ldots < t_r \leq t$ are the times at which pp i sends or receives a message, $m_1, m_2, \ldots$ are the corresponding message values, and $l_1, l_2, \ldots$ are the corresponding communication lines. We say that a history $h' =<(l'_1, t'_1, m'_1), \ldots, (l'_s, t'_s, m'_s)>$ is an *extension* of a history $h =<(l_1, t_1, m_1), \ldots, (l_r, t_r, m_r)>$, denoted by $h \leq h'$, iff $r \leq s$ and $(l_i, t_i, m_i) = (l'_i, t'_i, m'_i)$ for all $i = 1, 2, \ldots r$. For any pp i, let H_i denote its message history up to time Z.

We assume that if the message history h of a process i is known up to a time $t < Z$, then the next line l it wishes to communicate with is computable, say via a function $next\_line_i(h)$. If pp i does not wish to communicate with any line ever in the future, then $next\_line_i(h)$ will be NULL. Moreover, if l is an output line, then the corresponding message m and the time t when it is sent are also computable, say via a function $next\_out_i(h)$. (If the message is not sent by the time Z, then we need not know the actual values of t and m; thus in this case the values returned by the function $next\_out_i$ will be NULL.) Similarly, if l is an input line, then given the message m that arrived next along line l, and the time t when it was sent, we can compute the time t' that it will be received by pp i, say via a function $next\_in_i(h, l, t, m)$. (Again, if the message will be received at a time after Z, then the value t' returned by the function will be NULL.) Note that it is possible that pp i keeps waiting for an input along the line l forever, due to a permanent (acyclic) waiting in the physical system. The relationship between message histories H_i of the processes, and the above (computable) functions $next\_line_i$, $next\_out_i$, and $next\_in_i$, is stated more formally in [17].

Note that in the above formulation, the *instant* when a message is received could refer to the start of the receive operation, or to its end, etc. This detail is conveniently taken care of in the actual definitions of the specific functions $next\_in_i$, $next\_out_i$, and $next\_line_i$. Similarly, details such as the time taken by pp i to determine the next line to communicate with, the time taken by pp i to compute the next output, the time involved in executing a send or a receive operation, communication delays, etc. are conveniently buried into actual definitions of the above functions. In other words, our formulation remains valid irrespective of variations in such details.

3. ALGORITHM 1: A DISTRIBUTED SIMULATION ALGORITHM THAT TERMINATES NON-GRACEFULLY

In our distributed simulation algorithms, each physical system is simulated by a distributed simulator called a *logical system.* A logical system is a network of processes, called *logical processes* (or *lps* for short) that is topologically isomorphic to the physical system it simulates, with each pp being replaced and simulated by one lp. We assume that each communication line in the logical system is FIFO(first-in-first-out) with arbitrary but finite (possibly zero) communication delays, and error-free. Also, an lp can send a message on a line without having to wait for the receiver lp to be ready to receive it. Messages sent on a line are stored at the input port of the receiver lp on their arrival at the receiver lp, which later receives them from the input port, in a FIFO order for any given line.

A message m sent in the physical system on line (i,j) at time t is simulated in the logical system by a message (t,m) sent on line (i,j) at some point during simulation. Behavior of lp i closely resembles that of pp i. Tuple communications by lp i take place in the same order as the corresponding message communications by pp i. Below we give a pseudo code for any lp i in algorithm 1 at an abstract level. Each lp i maintains a variable S that refers to the message history of pp i that has been simulated so far in simulation.

```
(* Initialization *)
S := <> (* the empty message history *);
continue := TRUE;

while continue do
    begin
        (* Compute next line to communicate with. *)
        $l := next\_line_i(S)$;
        Cases:
            $l$=NULL: continue := FALSE;


            $l$=(i,j) for some j:
                begin
                    (t,m) := $next\_out_i(S)$;
                    if m=NULL then continue := FALSE
                    else begin
                            send the tuple (t,m) on line (i,j);
                            S := S // $(l, t, m)$;
                            if t=Z then continue := FALSE
                        end;
                end;


            $l$=(k,i) for some k:
                begin
                    wait until a tuple is available on line (k,i);
                    remove from input port (in FIFO order) the next
                    tuple (t,m) that arrived along the line (k,i);
                    t := $next\_in_i(S,l,t,m)$;
                    if t=NULL then continue := FALSE
                    else begin
                            S := S // $(l,t,m)$;
                            if t=Z then continue := FALSE
                      end
                end
        end (* of cases *)
    end; (* of while loop *)

(* continue = FALSE at this point. *)
Terminate (in a normal fashion).
```

4. PROOF OF CORRECTNESS OF ALGORITHM 1

In the following, we use S_i to refer to the variable S at lp i.

4.1. Safety Property Of Algorithm 1

Let us first state the safety property of algorithm 1 informally. At any point of simulation, for any lp i suppose $S_i = <(l_1, t_1, m_1), \ldots, (l_k, t_k, m_k)>$ for some $k \geq 0$. Then each of the following assertions holds.

1. $0 < t_1 < t_2 < \ldots < t_k \leq$ Z. Thus, send or receive operations of pp i are simulated in the chronological order of their time of occurrence.
2. For any tuple (l_s, t_s, m_s) in this sequence, message m_s was sent or received on line l by pp i at time t_s.
3. If a message m was sent or received by pp i on the line l at time t where $t \leq t_k$, then the tuple (l, t, m) is present in the above sequence. Thus, no message communication (i.e., a send or receive operation) by pp i up to time t_k is skipped in this sequence of tuples.

Theorem 1 below states the safety property of the algorithm more formally.

Theorem 1: At any time during simulation, for all initialized lps i: $S_i \leq H_i$.

Proof: The proof is by (strong) induction on the number of events in the logical system, where an event is the appending of a tuple to S at an lp, or the sending of a tuple by an lp, or the initialization of an lp. A detailed proof is given in [17].

4.2. Liveness Property Of Algorithm 1

At a point in simulation, we say that the *simulation computation has terminated* if: (i) every lp is either terminated or is waiting to receive a tuple, (ii) there are no tuples in transit, and (iii) if an lp is waiting to receive a tuple from an input line then no tuples corresponding to that line are available at the input port.

Theorem 2: Within a finite time from its start, the simulator computation reaches a state where the simulator computation is terminated, and for all lps i, $S_i = H_i$, i.e., all message communications among the pps up to time Z have already been simulated.

Proof: Follows from (A) and (B) below.

A. *Within a finite time simulator computation terminates:* The proof of this is straightforward; for details the reader is referred to [17].

B. *On termination of simulator computation— for all i, $S_i = H_i$* : To prove this, consider a moment when the simulator computation is terminated. Suppose there is at least one message communication up to time Z that has not been simulated yet, i.e., the corresponding tuple has not been appended to the corresponding message history S. Let $t < Z$ be the time of occurrence of the earliest send or receive operation not simulated yet. Let this operation take place at pp i on line l, and let the corresponding message value be m. By the safety property of algorithm 1, $S_i \leq H_i$ and by the above assumption, $S_i//(l,t,m) \leq H_i$.

By lemma 1 below, lp i is not terminated. Since the simulator computation is terminated, lp i must be waiting to receive an input. By lemma 2 below, lp i is waiting on line l, and the line l is an input line of lp i, say $l = (k,i)$. Suppose (l,t,m) above is the p^{th} tuple corresponding to line l in H_i. Let the p^{th} tuple corresponding to line l in H_k be (l,t',m).Consider the following two cases.

Case 1: *Suppose tuple (l,t',m) is present in S_k.* By the safety property, $S_k \leq H_k$. Therefore S_k has at least p tuples corresponding to line l. By the code, lp k has sent out at least p tuples on line l. But lp i has received only $(p-1)$ tuples on line l. Therefore the p^{th} tuple on line l is either in transit (which contradicts the terminated state of the simulator) or is at the input port of lp i (which contradicts the waiting state of lp i on the line l).

Case 2: *Suppose tuple (l,t',m) is not present in S_k.* Obviously, $t' \leq t$. Therefore by definition of t, $t' = t$ and $S_k//(l,t',m) \leq H_k$. By lemma 1, lp k is not terminated. Since the simulator computation is terminated, lp k must be waiting to receive an input. Since $S_k//(l,t',m) \leq H_k$, and l is an *output* line of lp k, lemma 2 is violated.

In both cases we get a contradiction. Hence the result follows.

Proofs of lemmas 1 and 2 below are given in [17].

Lemma 1: At any point in simulation, for any i:

lp i is terminated $\Rightarrow S_i = H_i$.

Lemma 2: At any point in simulation, for any i:

(lp i is waiting for input on line l, and $S_i \neq H_i$) $\Rightarrow$ ($\exists$ t and m such that $S_i//(l,t,m) \leq H_i$).

5. ALGORITHM 2: ENFORCING GRACEFUL TERMINATION

Note that when the simulation computation of algorithm 1 terminates, it is possible that some lps are still waiting to receive inputs and are not aware that the simulation computation has terminated. We will rectify this situation of "non-graceful termination" in this section. First, we show that at the point of termination of the simulation computation, the "wait for" relationship among lps is acyclic, i.e., there is no deadlock (a permanent cyclic waiting) in the simulator. Then we present a simple modification to the algorithm 1 to enforce a normal termination of each lp in an efficient manner.

Theorem 3: The simulator does not deadlock.

Proof: Suppose the simulation computation has terminated, and there is a cyclic waiting among lps. Based on this, we show below that the physical system deadlocks at some time.

This contradicts our assumption of no deadlocks in the physical system, and hence establishes the result.

Let (k,i) be any line on the above cycle in the logical system. Since $S_i \leq H_i$, t-value of $S_i < Z$, and $next\_line_i(S_i) = (k,i)$, pp i will start waiting to receive an input message on line (k,i) after executing the operation corresponding to the last tuple (if any) in S_i. Let $X = \{i \mid \text{lp } i \text{ is on the above cycle}\}$. For any $i \in X$, let $t_i =$ the t-value of S_i and let $t'_i =$ the above time when pp i starts waiting for input. Obviously, $t_i \leq t'_i$. We next show that for any $i \in X$, pp i will never receive a message after the time t_i. This will establish that this set of pps forms a deadlocked cycle at the time $T = max\{t'_i \mid i \in X\}$.

Suppose, to the contrary, some pp i, $i \in X$, receives a message after time t_i. Let u be the minimum value of any such time. Suppose the corresponding message m was sent on a line $l =$ (k,i) in the physical system at time v. Obviously, the line (k,i) is on the above cycle, and $k \in X$. Consider the following two cases.

Case 1. $v > t_k$: Then pp k received a message at a time w such that $t_k < w < v \leq u$. This contradicts the definition of u.

Case 2. $v \leq t_k$: Let (l, v, m) be the p^{th} tuple corresponding to line l in H_k. We derive a contradiction via the following steps.

1. The number of tuples sent by lp k on the line l

   $=$ the number of tuples corresponding to line l in S_k (follows from the code of algorithm 1)

   $=$ the number of tuples corresponding to line l in H_k up to time t_k (follows from the fact that $S_k \leq H_k$ and (t-value of S_k)$= t_k$)

   $\geq p$ (since $v \leq t_k$).
2. The number of tuples received by lp i along line l

   $=$ the number of tuples corresponding to line l in S_i (follows from the code of algorithm 1, and the fact that lp i is not terminated after receiving the last tuple along line l)

   $=$ the number of tuples corresponding to l in H_i up to time t_i (follows from the safety property of algorithm 1)

   $< p$ (by the definition of p, u, and the FIFO property of communication channels in the physical system).
3. From 1 and 2 above, the number of tuples received by lp i along line $l <$ the number of tuples sent by lp k along line l. This contradicts the fact that the simulation computation is terminated and lp i is waiting to receive an input from lp k. This completes the proof.

Since the wait-for graph is acyclic, a simple scheme (referred to as algorithm 2) can be used to avoid deadlock. An lp i, on deciding that it has completed its simulation, sends a special *termination message (NULL,NULL)* on all its output lines, then receives input messages until it has received one such message from each input line, and then terminates in a normal fashion. This guarantees that when the simulator computation terminates, each lp has terminated in a normal fashion, and also, all messages at the input port of an lp have indeed been received by it. An lp i decides that its simulation is completed when (i) the variable *continue* becomes false, or (ii) a tuple received by lp i from its input port has m-value NULL. A detailed code is given in [17].

Some of the above NULL messages can be avoided by sending (or waiting to receive) them on a line only if the previous tuple sent (or received) on the line had a t-value $< Z$ (as in [11,13,16]).

6. DISCUSSION

The main contribution of this paper is to identify a non-trivial class of systems that do allow distributed simulation with nearly zero overhead. This class of systems includes many cyclic systems. The essential characteristic of these systems is that a process at any time chooses to communicate with one specific line, and does not communicate with any other line until this communication is complete. Examples of such systems include several VLSI algorithms, distributed systems where processes exchange messages via a token that traverses the system along a predefined cycle, and queuing networks with no merge processes [3,4,16].

Other contributions of this work are discussed below.

Systems in this class have the special property that if the distributed simulation algorithm [4] is directly applied to these systems, then the overhead would be less than the usual overhead of [4] in simulating more general classes of systems. In general, the algorithm [4] behaves as a *multi-phase* algorithm. In each phase, first termination of the primary computation is detected. Then, from the states of the various processes in the simulator, the algorithm computes the next event to be simulated. If indeed such an event exists, then the primary computation is resumed by simulating this event. Along the lines of our proof of theorem 2, one can show that if this distributed simulation algorithm is directly applied to our systems, then the following holds — whenever the primary computation terminates, it is guaranteed that all events have already been simulated, i.e., there will not be any next event to be simulated, and therefore the primary computation won't be resumed. Hence, termination occurs only once during the simulation, and thus the time lag between occurrence of termination and its detection, and the time to compute the next event, will be wasted *only once* in the simulation.

Algorithm 2 can be viewed as a simple modification of the scheme in [4], in order to further reduce overhead in simulating this class of systems. First, we eliminate the computation of next event to be simulated after termination detection, since it is unnecessary, as observed above. Next, along the lines of our proof of theorem 3, we observe that when the primary computation terminates, the "wait-for graph" among the lps in the simulator will be acyclic; obviously, such a waiting can be avoided by using termination messages [represented by (NULL,NULL) in algorithm 2].

If the algorithm in [3] is applied to our systems, a large number of overhead NULL messages would be generated. Almost all these messages are useless — they are not necessary for the correct operation of the simulator (as shown by correctness of our algorithm 2), and they don't speed up simulation since receiving a (t,NULL) message with $t < Z$ will not be helpful to lp i in generating non-NULL output tuples; after receiving such a message, lp i would still keep on waiting for the next input message on the same line, possibly generating only some more output NULL messages. The only useful NULL message from a process to another is one that carries information that all events between them up to time Z have already been simulated. Our algorithm 2 can be viewed as a much simplified version of [3], with this observation.

Acknowledgements

The basic ideas of this paper evolved out of discussions with Professor M. Gouda of University of Texas, Austin. I appreciate his suggestions for improvements to several earlier drafts of this paper. I am thankful to Professor J. Misra for many fruitful discussions with him on the subject of distributed simulation, and for his financial support for this work.

REFERENCES

1. Bezivin, J. and Imbert, H. Adapting A Simulation Language To A Distributed Environment. *Proceedings of the 3rd International Conference on Distributed Computing Systems*, 596-603, Ft. Lauderdale, Florida, 1983.

2. Chandy, K. M., Holmes, V., and Misra, J. Distributed Simulation Of Networks. *Computer Networks.* **3,** 1 (Feb. 1979), 105-113.

3. Chandy, K. M. and Misra, J. Distributed Simulation: A Case Study In Design And Verification of Distributed Programs. *IEEE Transactions on Software Engg.* **SE-5,** 5 (September 1979), 440-452.

4. Chandy, K. M. and Misra, J. Asynchronous Distributed Simulation Via a Sequence of Parallel Computations. *Communications of the ACM.* **24,** 4, (April 1981) 198-205.

5. Christopher, T., et. al. Structure of a Distributed Simulation System. in *Proceedings of the 3rd International Conference on Distributed Systems*, Ft. Lauderdale, Florida, 1983.

6. Davidson, D. and Reynolds, P. Performance Analysis of Distributed Simulation Based on Active Logical Processes. *1983 Winter Simulation Conference Proceedings*, 267-268, Dec. 12-14, 1983.

7. Dijkstra, E. W. and Scholten, C. S. Termination Detection for Diffusing Computations. *Information Processing Letters.* **11,** 1 (August 1980). Also, *EWD article,* (January 1979).

8. Hoare, C. A. R. Communicating Sequential Processes. *Communications of the ACM.* **21,** 8 (August 1978), 666-677.

9. Jefferson, D. R. and Sowizral, H. A. Fast Concurrent Simulation Using The Time Warp Mechanism, Part I: Local Control. Technical Report, The Rand Corporation, Santa Monica, California, December 1982.

10. Jefferson, D. R. Virtual Time. *ACM Transactions on Programming Languages and Systems.* **7,** 3 (July 1985), 404-425.

11. Kumar, D. Design and Performance Analysis of a Distributed Simulation Scheme. *CMG '85 International Conference*, Dallas, Texas, December 10-13, 1985.

12. Kumar, D. A Class of Termination Detection Algorithms For Distributed Computations. *Fifth Conference on Foundations of Software Technology & Theoretical Computer Science*, New Delhi, India, December 16-18, 1985, Lecture Notes in Computer Science, #206, pp. 73-100, Springer-Verlag; also Technical Report TR 85-07, Dept. of Computer Sciences, University of Texas at Austin, May 1985.

13. Kumar, D. Simulating Feedforward Systems Using a Network of Processors. *Nineteenth Annual Simulation Symposium*, Tampa, Florida, March 12-14, 1986.

14. Kumar, D. A Novel Approach To Sequential Simulation. *IEEE Software*, September 1986.

15. Kumar, D. Efficient Algorithms For Distributed Simulation And Related Problems. Ph.D. Dissertation, Department of Computer Sciences, University of Texas, Austin, May 1987.

16. Kumar, D. A Low Overhead Distributed Simulation Scheme And Its Performance Evaluation. Technical Report CES-88-09, Department of Computer Engg., Case Western Reserve University, May 1988.

17. Kumar, D. A Class of Systems Permitting Low Overhead Distributed Simulation. Technical Report CES-88-13, Department of Computer Engg., Case Western Reserve University, July 7, 1988.

18. Lamport, L. Time, Clocks, and the Ordering of Events in a Distributed System. *Communications of the ACM.* **21**, 7 (July 1978).

19. Lavenberg, S., Muntz, R., and Samadi, B. Performance Analysis of Distributed Simulation Using Rollback. *Proceedings of the Fifth Annual Conference on Performance*, 1983.

20. Misra, J. and Chandy, K. M., Proofs of Networks of Processes. *IEEE Transactions on Software Engineering*, **SE-7**, 4 (July 1981), 417-426.

21. Misra, J. Distributed Discrete-Event Simulation. *ACM Computing Surveys.* **18**, 1 (March 1986), 39-65.

22. Owicki, S. and Gries, D. An Axiomatic Proof Technique for Parallel Programs. *Acta Informatica.* **6**, (1976), 319-340.

23. Peacock, J. K., Wong, J. W., and Manning, E. G. Distributed Simulation Using A Network of Processors. *Computer Networks.* **3**, 1 (Feb. 1979), 44-56.

24. Peacock, J. K., Manning, E. G., and Wong, J. W. Synchronization of Distributed Simulation Using Broadcast Algorithms. *Computer Networks.* **4**, (1980), 3-10; also Peacock, Wong and Manning, *Proc. of Winter Simulation Conference*, December 1979.

25. Reed. D. A. A Simulation Study of Multimicrocomputer Networks. *Proceedings of International Conf. on Parallel Processing*, 161-163, August 1983.

26. Reed, D. A., Malony, A. D., and McCredie, B. D. Parallel Discrete Event Simulation Using Shared Memory. *IEEE Transactions on Software Engineering.* **14**, 4 (April 1988).

27. Reynolds, P. A Shared Resource Algorithm for Distributed Simulation. in *Proceedings of the 9th International IEEE Architecture Conference*, Austin, Texas April 1982.

28. Sauer, C. H. and MacNair, E. A. Simulation of Computer Communication Systems. Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632, 1983.

29. Seethalakshmi, M. A Study And Analysis of Performance of Distributed Simulation. Master's Report, Dept. of Computer Sciences, University of Texas, Austin, Texas 78712, May 1979.

Distributed Computation on Concurrent Processors
Time Shared Multicomputing

Vijay K. Madisetti and **David G. Messerschmitt**

Electrical Engineering and Computer Sciences.
University of California at Berkeley,
Berkeley, CA 94720

Abstract

In this paper, we propose models for distributed computation on concurrent multiple processor systems. We propose rigorous paradigms for asynchronous distributed computation and analyze their properties using probabilistic methods. Realizable distributed computing constructs are then formulated. Algorithms for optimizing computing resource allocation are presented which enable multiprogrammed processors to cooperate efficiently in a medium-grained computation model. A concurrent system programming language, COSPROL, is introduced. Attractiveness of COSPROL results from the fact that very few assumptions other than observable input-output descriptions are used to optimize performance.

1. Introduction

Distributed computation on concurrent multiple processing systems has been the focus of much interest in the past few years. Supercomputing performance is achieved for certain classes of problems at a much reduced cost. A number of interesting models have been proposed in literature to model such systems. A rigorous analysis of the properties of such models has not been vigorously pursued and with the increasing availability of commercial multiple processor systems such studies have gained importance.

In this paper, we propose a rigorous model for distributed computation on multiprogrammed, multitasking multiple processor systems. The models and their constructs are quantitatively evaluated using probabilistic arguments, and their properties are discussed. Our objective is to arrive at a machine-independent specification of a distributed asynchronous computation application, which optimizes throughput solely on the basis of the input-output properties of the program segments, which are transparent to the programmer. These algorithms would also be applicable to the case when a uniprocessor workstation is upgraded to the multiple processor environment. The problems addressed by such computing architectures are computation intensive (300+ MFLOPS), data intensive (Gbytes) and demand a high I/O data transfer (Mbytes/s) capability.

2. Models for Distributed Computation

In this architecture, a number of independent processors, each with its own local memory, are connected together by an interconnection network. The processors are multitasking and multiprogrammed and use messages to communicate with each other. Our interest lies primarily in the medium grained model of computation, in which a number of processors are assigned jobs in parallel, realizing the concurrent implementation of a particular algorithm that has a medium to coarse granularity of tasks. Communication is expensive in these architectures. However, a rich interconnectivity is usually provided allowing the network of processors to represent a wide variety of topologies.

There are a number of interesting issues to be considered when dealing with time shared multicomputers. Most present operating environments advocate a *space shared* approach vis-a-vis a *time shared* approach to multicomputing (see Athas and Seitz, 1988). In a space shared approach, users are assigned non-overlapping sets of processors for their requirements. The advantages of using this approach are the (typically) predictable run times, and less overhead in balancing load across a number of independent processes. However, the efficiency of the computing nodes is then dependent on the user program and task assignment and load balancing routines lack the flexibility available in a time shared multicomputing environment. This paper primarily deals with algorithms for time shared multicomputer systems, when algorithms assigning resources to a

---

This research was supported in part by the Computer Sciences Division, Shell Development Co., Houston, TX. and in part by the California Microelectronics Research Foundation (MICRO).

particular application must take into account the requirements of other concurrent programs on the same distributed computing system. We first describe the constructs for modeling distributed computation, and then we study some properties of these constructs.

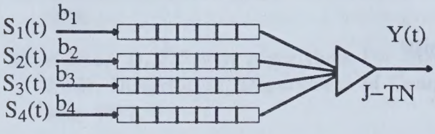


Figure 1

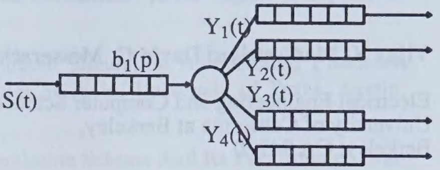


Figure 2

A system programming language, COSPROL, is developed allowing a high level description of a concurrent algorithm, independent of any particular concurrent computer system. Some algorithms are then presented through which a COSPROL preprocessor assigns subtasks to different nodes. Efficient allocation of resources is considered and some new schemes for such allocation are presented.

2.1 Distributed Computation Constructs

We now propose a formal definition of the models.

A Join-Type Network (J-TN)

Consider the J-TN illustrated by Figure 1. Sources $S_1(t), S_2(t), \cdots, S_N(t)$ provide N inputs to their respective buffers $b_1, b_2, b_3, \cdots, b_N$. The sources S_i transmit at rates λ_i, with the p^{th} arrival at time $t^{S_i}_p$ to the i^{th} buffer. We quantify our model of the J-TN as follows,

$$b_i(p) = b_i(t^{S_i}_p)\cdot I\{t^{S_i}_{p+1} > t \ge t^{S_i}_p\}$$

$$Y(t+\theta) = \sum_{p=0}^{\infty}(f(b_1(p), b_2(p), \cdots, b_N(N))\cdot I\{t^*_{p+1} > t \ge t^*_p\}$$

where

$$t^*_p = \{\min t \mid b_i(p) \ne \phi, i \in (1, N)\}$$

and $f(b_1(p), b_2(p), \cdots, b_N(p))$ is a program which operates on the data in the buffers and I is the indicator function. Qualitatively, the output is produced only when all the input buffers have atleast one input, such that $Y(p) = f(b_1(p), b_2(p), \cdots, b_N(p))$. θ is the processing time required by the J-TN to generate an output given the inputs. The standard multiplexor is a degenerate example of a J-TN.

A Fork-Type Network (F-TN)

A Fork-Type Network is illustrated in Figure 2. A F-TN has a single input buffer $b_1(p)$ and N outputs $Y_i, i = 1,2,\cdot,N$ and the outputs are related to the inputs according to

$$Y_i(t) = \sum_{p=1}^{\infty} f(b_1(p))\cdot I\{t^*_{p+1} > t \ge t^*_p\}$$

In general, we could provide for a *selector set* S which determines which of the outputs $Y_i(t)$ would change, allowing a probabilistic routing, varying with time, from the inputs to the outputs. A standard demultiplexor is then a special case of the the F-TN, where each successive input (in time) is routed to a successive output link (in space).

A Simple Process

A simple process p is one which accepts an input, $x(n)$, and produces an output, $y(n)$. If P is the processor which assigns c_p fraction of its CPU time to p then t_p is defined to be the time taken by p to produce an output. A simple process communicates with other processes only when it has completely processed the data assigned to it. (Figure 3).

A Compound Process

A compound process is composed of a number of processes interacting with each other. Each of these processes is spawned by an independent multitasking processor which assigns a certain fraction of its CPU time to its child process. An example of a compound process is shown in Figure 4 where processes $p_1, p_2, \cdots, p_n$ interact to process input data. The data flow allows for feedbacks, and multiple subsequent visits by the input to each process. The processes may generate shared variables which are transparent to the input-output specifications. We will discuss these compound processes further and develop their properties. Most multicomputing environments can be described with compound processes.

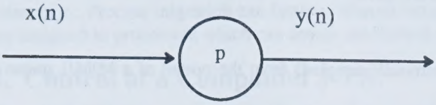


Figure 3

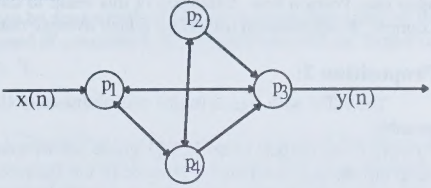


Figure 4

A Simple J-TN (SJ-TN)

A Simple J-TN is a J-TN whose inputs are all simple processes. The SJ-TN processes the data from each of the individual processes to produce an output according to the equations for the SJ-TN.

A Compound J-TN (CJ-TN)

A Compound J-TN is one which has atleast one input which is a compound process.

Pipes

Pipes allow for processes to communicate in a specified order. Output from one process is fed into the input of another process through a pipe.

At this point we digress and develop a few mathematical properties of the J-TN which will be useful in subsequent analysis of distributed computation systems. In effect, we will be deducing some stability results which will help us determine which models for computation are realizable in practice.

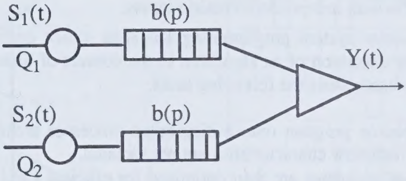


Figure 5

2.2 Properties of Constructs

Stability: Consider a J-TN with two input sources, S_1 and S_2 as shown in Figure 5. If the buffer sizes in b_1 and b_2 are given by $|b_1|$ and $|b_2|$ then a J-TN is stable if and only if $|b_1| - |b_2| < M$ for finite M as $t \rightarrow \infty$. Therefore, a J-TN is unstable if its buffer size grows without bound.

Proposition 1:

Let the sources to a J-TN be S_1 and S_2. If the sources are independent and Poisson, then the J-TN is unstable for all positive rates $\lambda_i > 0$.

Proof: Let us assume that the Queues Q_1 and Q_2 have service rates, μ_1 and μ_2 and let the inputs S_1 and S_2 be independent Poisson sources.
Case 1: $\mu_1 = \mu_2 = \infty$ and $\lambda_1 = \lambda_2 = \lambda$

The proof is as follows: If b_1 and b_2 are the buffers from sources 1 and 2, then $|b_1| - |b_2|$ follows a one dimensional random walk with the following transition probabilities.

$$P_{k,k\pm 1} = \frac{1}{2}$$

The underlying Markov chain is, therefore, **null recurrent** This also implies that $|b_1| - |b_2|$ grows without bound for all positive rates. (The probability of $|b_1| - |b_2|$ becoming zero is one, however, the expected time for this to happen is infinite.)
Case 2: λ_1 is not equal to λ_2

In this case, the underlying Markov chain is **transient** and hence, the J-TN is unstable.
Case 3: If μ_1 and μ_2 are $< \infty$ then by *convexity* arguments, the instability result still holds, though there is latency in the output.

Therefore, we can conclude that independence of the two Poisson inputs is a sufficient condition for instability.

This (surprisingly) strong new result was derived using simple probability with very few assumptions other than Poisson flow. Extension of this result to other general flows is under study. Deterministic flows, for example, do not result in instability if their average rates are same.

Proposition 2:

The J-TN with inputs routed probabilistically (Bernoulli sampled) from the output of a M/M/1 queue is unstable.

Proof: If the outputs of an M/M/1 queue are Bernoulli sampled and fed to the inputs to the J-TN, (from the sampling theorem for Poisson processes), the sampled output processes are independent and from Proposition 1, this implies instability.

Theorem 1:

The J-TN fed by N sources with rates λ_i is unstable for all positive rates if any two inputs are mutually independent.

Proof: Follows from Propositions 1 (and 2).

2.3 Discussion on Join-Type Networks

Theorem 1, implies that a J-TN is not *realizable* in practice with finite buffer sizes if its inputs were independent and Poisson. It is conjectured that the J-TN is unstable for all random inputs with independent inter-arrival times (independent increments). This can be visualized by a simple example. Consider a simple J-TN with two inputs. The interarrival probability density function has mean a and variance σ. A sequence of arrivals can then be constructed, $(a-\sigma)$, $2(a-\sigma)$ ..... from input 1 and a sequence of arrivals at times $(a+\sigma)$, $2(a+\sigma)$ .....,from input 2, which leads to unbounded buffer sizes. It would, therefore, be impossible to assign buffer sizes for J-TNs with independent Poisson inputs.

COSPROL is a concurrent system programming language which enables the algorithm developer to define a parallel paradigm for execution of an algorithm in the context of time shared multicomputer environment. The COSPROL preprocessor, does the following tasks.

1. It maps the COSPROL source program onto a distributed processor architecture, such that the algorithm constraints, precedences and data flow characteristics are not violated.
2. The Join-Type Networks of processes are then optimized for efficient CPU utilization, as will be described in the following sections.
3. By adaptive perturbation techniques, the COSPROL distributed architecture consisting of asynchronously executing processors is synchronized.
4. By keeping a current description of the critical sections of the network, the throughput can be dynamically varied with fluctuating load.

The individual processes are pieces of code which are defined by the programmer. COSPROL optimizes resource allocation solely on the basis of input-output characteristics of the output flow. The execution times, process fractions, and communication protocols within processes are not assumed to be known beforehand, making such a description of concurrent programs very attractive.

3. Control of Simple J-TN of Processes

A J-TN consists of a join of the outputs of a number (say, N) of simple processes. Each of these processes, p , is spawned by a processor which assigns a fraction of its computing time, c_p, to it. Heavily utilized processors can assign small c_p, if at all, while lightly loaded processors can devote a higher fraction of their computing time to processes. In a time shared multicomputing environment, the number of processes per processor is also variable. An efficient assignment algorithm would have the capability to track such variations in load in the context of the concurrent program. If the constituent processes (in a J-TN) produced outputs at varying rates, buffering would be difficult, and efficiency would be lost due to enforced idle times. The algorithm for resource allocation in a J-TN is straightforward. We will briefly outline it.

Let us assume that the process p, consumes c_p and takes a processing time t_p to complete its task. Let $T = (t_1, t_2, t_3, \cdot, t_p, \cdot, t_N)$, and $C = (c_1, c_2, \cdot, c_p, \cdot, c_N)$ and $\Lambda = (\alpha_1, \alpha_2, \cdot, \alpha_p, \cdot, \alpha_N)$, then

$$\Lambda = C \cdot T$$

We now define $\rho = \max_k(\frac{\alpha_k}{c_k})$ and the reassigned processor fractions , c_p^{new}, for each process will then be given by

$$c_p^{new} = \frac{\alpha_p}{\rho}$$

ρ keeps track of the "bottleneck" process, and the J-TN can be dynamically scaled up or down depending on the value of ρ. Process migration can further enhance the speed of completion, if the processes with the largest α_j are assigned to processors which can assign the highest c_j 's.

4. Control of a Compound J-TN.

In this section, we consider the case of a general J-TN, dealing with the join of compound processes. A *compound* process is defined as one having a number of cooperating and concurrently executing processes. Regardless of the interconnection, the output of the compound process is joined with the output of other compound processes in a compound J-TN (CJ-TN). The CT-JN allows us to consider medium-grained model of computation useful for most scientific applications. We assume no specific knowledge of the tasks carried out by each process is available to the CJ-TN, other than the input/output constraints and task completion times. Our proposed stabilization algorithm is independent of the programs executed subject only to some assumptions which will be described below. Each of the sub-processes p executes a different task and the dependencies between the sub-processes make calculations for optimal CPU allocation intractable.

Our algorithm is best described by the *Activity Plot* Figure 6, which plots the process activity versus time. The shaded regions indicate when a process is active. The dependencies are depicted by arrows. In principle, the active regions of a compound process are not observable. The plot for the CJ-TN is plotted for the case when each process has been assigned equal CPU fraction c .

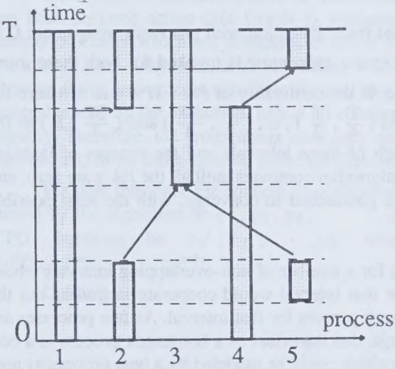
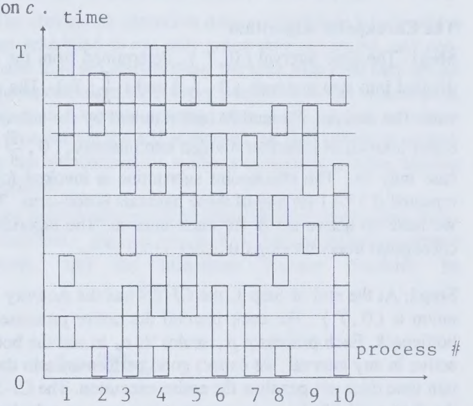


Figure 6



Graph 1

We analyze the performance of our algorithms for a simulated activity plot shown in Graphs 1-5. The test pattern was generated using a random number generator. With a probability 1/2, the active regions are extended into the next checkpoint interval. The process fractions are assigned from a uniform distribution $U(0,1)$. The successive graphs depict the performance on a ten processor system, with and without process migration and with and without checkpoint optimization. The execution times are assumed deterministic for the moment.

4.1 Optimization Algorithm for Compound Process

We now describe the *checkpoint* algorithm which optimizes the CPU assignment amongst the processes constituting a compound process. Each process while executing its assigned task, communicates with other processes within the same compound process. Each process (within the compound process) is assigned a fraction of CPU time by its parent processor depending on its multiprogramming requirements. Our algorithm is an adaptive one, and involves invokes a subroutine called the *checkpoint* The net outcome of the checkpoint algorithm is a list of *active* processes for nonoverlapping intervals of time $(t_i\ ,t_{i+1})$, $A_{t_i\ ,t_{i+1}}$ for $i = 1\ ,2\ ,3\ ,\cdots\ ,Z$ such that the union of the intervals is $(0\ ,T)$, where T is the total time required for execution. We now give the pseudocode description of the subroutine.

```
subroutine checkpoint ( $q_1 ,q_2 ,A\ ,IA\ ,PA\ ,\varepsilon\ ,T$ )
```

/* The initial processor assignment is $\underline{C} = (c_1 , c_2 , . . , c_N)$

/* Defining the operator $\nu_p \underline{C}$ as follows*/

$$\nu_p . \underline{C} = (c_1 , c_2 , \cdot , (1+\sigma)c_p , \cdot \cdot , c_N) , \text{ for } t \in (q_1 , q_2) \cap (0 , T)$$

$$= \underline{C} , \textit{ otherwise}$$

$$\Delta Q_p = \alpha \nu_p \underline{C} - \alpha \underline{C}$$

$$\Delta Q = q_2 - q_1$$

$$\gamma_p = \frac{\Delta Q_p}{\Delta Q}$$

Each process , p , is assigned to the set of active states A_{q_1,q_2} , the set of inactive states IA , or the set of partially active states , PA , according to the following rule:

$$\sigma - \varepsilon < \gamma_p \leq \sigma , p \in A_{q_1,q_2}$$

$$\varepsilon < \gamma_p \leq \sigma - \varepsilon , p \in PA$$

$$0 \leq \gamma_p < \varepsilon , p \in IA$$

The Checkpoint Algorithm

Step1: The time interval $(0 , T)$, determined from the initial trace execution with processor assignment C is divided into two intervals $(0 , \frac{T}{2})$ and $(\frac{T}{2} , T)$. The *checkpoint subroutine* is invoked for both these intervals. The sets A , PA and IA are returned by the subroutine. If the cardinality of PA , $| PA |$ is non-zero for either interval, it is further divided into intervals, $(0 , \frac{T}{4})$ and $(\frac{T}{4} , \frac{T}{2})$, or, $(\frac{T}{2} , \frac{3T}{4})$ and $(\frac{3T}{4} , T)$ as the case may be. The checkpoint subroutine is invoked for each of these intervals and the process of iteration repeated if $| PA |$ for any of these intervals is non-zero. The algorithm continues until all the PA s are zero, and we have an active set A for each interval. The algorithm is guaranteed to converge, with the least possible checkpoint interval being $(0 , \varepsilon \Delta Q)$.

Step2: At the end of Step 1, the CJ-TN has the Activity Sets for a number of non-overlapping intervals whose union is $(0 , T)$. For each interval the *active* processes for that interval would cooperate in finding out the bottleneck. Each processor p, scales its c_p to suit the bottleneck process for that interval. As few processes are active in any interval, we expect good performance in the sense that the effect of a bottleneck process at a certain time does not penalize the entire execution. The CJ-TN (which could be modeled by a host processor) uses the data available from the parent processors, to calculate the new assignments. Once the efficient assignment of CPU to the compound process in a CJ-TN is completed, the entire CJ-TN is controlled using the scaling algorithm used for a Simple J-TN.

Proposition 1:

Let the events executed by the active processes in the interval (q_2 , q_1) be J_{q_2,q_1} . The time required for processing these events is the $f \ (J_{q_2,q_1} , \underline{C}) = q_2 - q_1$, where $\underline{C}$ is the resource allocation vector and $f : J_{q_2,q_1} X \ R \rightarrow R$. Let $\underline{\alpha} = (\alpha_1 , \alpha_2 , \cdot , \alpha_N)$, then a sufficient condition for

$$f \ (J_{q_2,q_1} , \underline{\alpha . C}) < f \ (J_{q_2,q_1} , \underline{C})$$

is that $\alpha_i > 1$ for all i .

Justification:

The proof of sufficiency is immediate. However, the fact that it is not a necessary condition may not be obvious.

Let $p_1 , p_2 , \cdot \cdot , p_5$ execute events J_{q_2,q_1} in the interval (q_2 , q_1). Let the processes p_1 , p_2 , p_3 , p_4 execute events $J_{q_2,q_1} - E$. E is the event assigned to process p_5 which records the progress of the execution of events among the other processes. Therefore any vector $\underline{\alpha}$ with $\alpha_i > 1$ for $i \in (1,4)$ and $\alpha_5 = 1$ will lead to a smaller $f\ (J_{q_2,q_1}, \underline{C})$.

Proposition 2:

An event a which belongs to the set of inactive processes for an interval (q_2, q_1) can be scheduled to occur independently of other events in the interval which are associated with processes that are active, i.e. $p \in A_{q_2,q_1}$.

Justification:

If an event a which is scheduled by an inactive process could affect an event scheduled by an active process, then perturbing the service rate of the inactive process, would affect the total execution time. Hence, the events associated with inactive processes are independent of other active processes during the interval (q_2, q_1). An important consequence is that inactive processes can be scheduled whenever resources are available later or when they turn active.

Theorem 2:

Every process p, when active for an interval (q_1, q_2), $p \in A_{q_1,q_2}$, is on the critical path for that interval.

Justification:

By construction, a process is classified as active if perturbation of its assigned fraction results in a perturbation of a like amount in the total execution time. Therefore *every* process is on the critical path whenever it is active. It shares the critical path with all other processes that are also active within the same interval.

This is one of the main results of this section. The checkpoint algorithm does not complete a task until it has been declared active (See Graph 1). Processes are scheduled to run only when they are active, and concurrent processes which are scheduled to run at the same time are shifted along the time axis until they are on the critical path. Therefore, after scheduling with a checkpoint algorithm, every process is on the critical path, slowing down one process can slow the total execution time. Likewise speeding up a single process cannot, in general, speed up the execution, unless all concurrent processes in the active interval are also scaled up proportionally. Therefore, the programmer need only specify the interconnection within a compound process, leaving optimal allocation and execution to the checkpoint algorithm.

We now introduce the concept of nonhomogeneity in multiprocessing systems. Let the checkpoints determined by the algorithm be $q_1, q_2, q_3, \cdot, q_M$ such that $\bigcup(q_{i+1},q_i) = (0, T)$. Let the initially assigned CPU fractions be $c_1^I, c_2^I, \cdot, c_N^I$ respectively, and the maximum fraction available be $c_1^{\max}, c_2^{\max}, \cdot, c_N^{\max}$.

Defining

$$\kappa_p = \frac{c_p^{\ I}}{c_p^{\ \max}} \quad , \quad \kappa_p^{\ *} = \max_p \kappa_p$$

$$\kappa_{p \in A_{q_i,q_i}} = \frac{c^I{}_{p \in A_{q_i,q_i}}}{c^{\max}{}_{p \in A_{q_i,q_i}}} \quad , \quad \kappa^*{}_{p \in A_{q_i,q_i}} = \max_p \kappa_{p \in A_{q_i,q_i}}$$

If γ denotes the ratio of the change in total execution time after the checkpoint algorithm to the unoptimized time T then

$$\gamma = \frac{\sum_{i=1}^{M}(q_{i+1} - q_i)(\kappa^*{}_p - \kappa^*{}_{p \in A_{q_i,q_i}})}{\sum_{i=1}^{M}(q_{i+1} - q_i)\kappa^*{}_p}$$

γ is defined as the **degree of nonhomogeneity** and it is an indicator of the extent of the disparity between the tasks and the process CPU assignments. The larger γ is, the greater the gains from our optimization algorithms, since a smaller γ would imply a homogeneous partitioning of tasks before preprocessing with COSPROL. In some cases, calculation of allocation may be simplified if all the $c^I{}_p$ were equal for all p. It must be noted here that the checkpoint algorithm extracts enough information from the constituent processes in the concurrent program to determine the processes which are active for a large number of subintervals. Obviously these processes need the largest fractions of CPU and a suitable scheme can be devised which reassigns processes to processors such that the computation intensive processes get assigned to processes with larger $c^{\max}{}_p$. (See Graph 2-3).

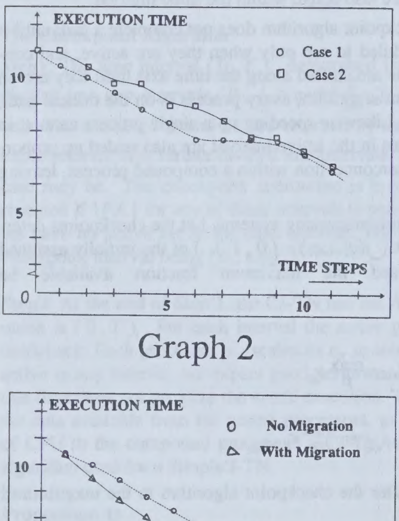
4.2 Process Migration

In this section, we examine some strategies for migrating processes to other processors in the system. Consider N processes $1, 2, \ , p, \ , N$ spawned by processors $P_1, P_2, \cdot, P_N$ each with a maximum available fraction of CPU $c^{max}_{P_1}, c^{max}_{P_2}, \cdots, c^{max}_{P_N}$. The checkpoint algorithm generates sets of active processes $A_{q_i, q_{i+1}}$ for intervals (q_i, q_{i+1}) with an initial assignment of CPU fractions equal to the maximum available per processor. Let the cardinality of the active sets $|A_{q_i, q_{i+1}}| = Z_i$, and the processors be ordered in the decreasing fraction of CPU available, $P_1, P_2, P_3, \cdot, P_N$. We consider two cases for process migration which enhance processor utilization.

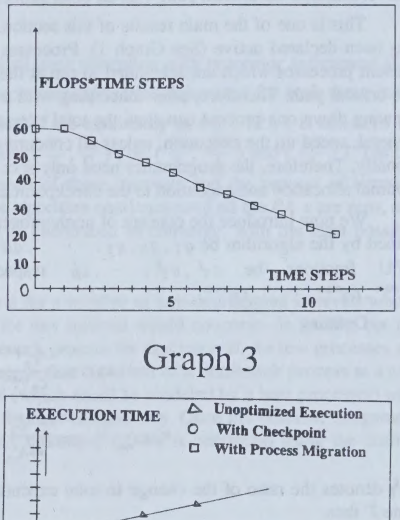
Case 1: Process migration

At the end of the checkpoint algorithm, each of the processes belonging to the active set $A_{q_i, q_{i+1}}$ is reassigned to one of the processors $P_1, \cdot\cdot, P_{Z_i}$. Each process then has its CPU assignment scaled up by $\frac{c^{max}_{P_y}}{c^{max}_{x}}$, where y and x are chosen such that the ratio is the minimum, say β_i^{-1}, for that interval. In practice we can scale up the CPU assignment by a larger fraction but this requires a larger amount of computation to ensure that each process is assigned to a processor which can scale its CPU fraction by the same amount. The speedup factor γ can then be represented as

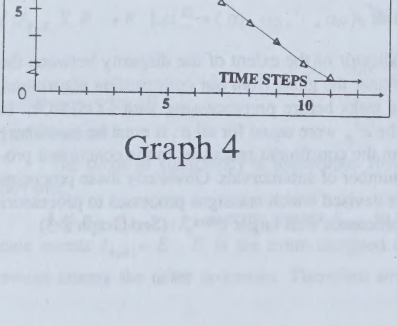
$$\gamma = \frac{\sum_{i=1}^{M}(q_i - q_{i+1})\kappa^*_p - \sum_{i=1}^{M}(q_i - q_{i+1})\beta_i + Z_i \cdot C}{\sum_{i=1}^{M}(q_i - q_{i+1})k^*_p}$$



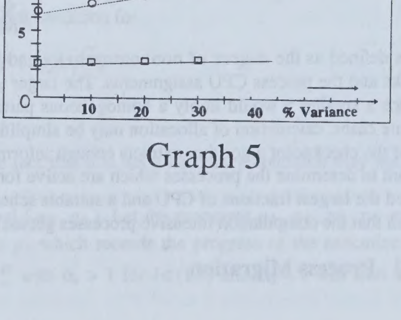
Graph 2



Graph 3



Graph 4



Graph 5

Since $\beta_i \leq \kappa^*_{p \in A_{q_i, q_{i+1}}}$ process migration can result in improvement in speed if the overhead , $Z_i \cdot C$, associated with process migration is assumed negligible. This is only justified if the intervals (q_i , q_{i+1}) are large compared to the costs of context switching and communications associated with process migration. Therefore, the processes are migrated to available processors with larger $c^{max}{}_p$ if and only if the interval (q_i , q_{i+1}) is sufficiently large.(See Graph 4). This overhead is significant only for the first few data samples consumed by the CJ-TN. The evaluation of A and IA at each step, allows the reduction of the complexity of the algorithm at each successive subinterval. This is because of the fact that if a process is active during an interval it maintains that state for all subintervals.

Case 2: Data-dependency:

In the above analysis, we have tacitly assumed that the data dependency of the process segments within the checkpoints can be expressed as some function of ε. The efficiency of the checkpoint algorithm is then a function of ε. As discussed above, the checkpoint algorithm moves process segments on to the critical path, however processing them *earlier* has no effect on the total run length. A **monitor** on each processor keeps a list of process segments which can be executed over a number of checkpoint intervals without affecting the total execution time. Consequently, whenever, a processor completes execution of its program within a checkpoint interval much ahead of other processors, it processes one of these *floating* off-critical path tasks.

This modification boosts the efficiency at the cost of a floating queue of tasks which can be executed with some latitude in the time of their completion (See Graph 5).

5. COSPROL : Rules and Syntax

We now briefly describe the COSPROL syntax and rules, which enable a concurrent system developer to specify an algorithm for implementation on a distributed computing architecture. The algorithm specified in COSPROL can then be analyzed by a preprocessor to allocate the architecture, assign resources, and optimize performance. We will first describe the syntax and then discuss the processing algorithms used by the COSPROL preprocessor. Paucity of space forces us to be content with a very simplified description.

Syntax

In the following discussion we denote processes by capital letters A , B , C and so forth. J-TN denotes a Join-Type Network. Unless otherwise noted we assume each process is spawned by separate parent processor.

a. A = SP (A)
b. X = CP [AB, AC, BC, CD]
c. S1 = SJ-TN [z_1 *A ,* z_2 *B]*
d. C2 = CJ-TN [z_1 *A ,* z_2 *B ,* z_3 *X]*
e. C3 = n CJ-TN [z_1 *A,* z_2 *B, C2]*
f. C5 = F-TN [A , S , z_1*]*
g. C5 / C4

(a) defines A as a simple process. *X* is a compound process, comprising of a cluster of processes *A, B, C* and *D* processes *A* and *B* , *A* and *C* etc communicating with each other. S1 is a Simple J-TN of z_1 processes described by A and z_2 processes described by B . C2 is defined as a Compound J-TN of processes, A , B and X of which atleast one is a compound process. C3 is defined to be a CJ-TN of order **n** which implies that it is the J-TN of at least one process of order **n - 1.** For example if n = 1 then it is the CJ-TN, if however, n = 2 then, it is second order hierarchical CJ-TN of first order CJ-TNs. C5 is a Fork-Type Process, with A as the input process, *S* is the selector process which determines which of the outputs are available to C5 and z_1 is the maximum number of output buffers. (g) defines the sequential connection of two processes C5 and C4. The output of C5 is piped into the input of process C4. This definition is similar to the UNIX pipes. We now present an example to illustrate use of COSPROL to specify distributed computation algorithms.

5.1 Examples

Example 1: Phase Shift Migration

The COSPROL representation of phase shift [1] migration can be elegantly described as follows.

Seismic Migration Algorithm

S1 = SJ-TN [a FFT]
C2 = S1 / A1

P1 = F-TN [C2 , ALL , a]
C3 = SJ-TN [a P1 | FFT , E]
C4 = C3 | M
P2 = F-TN [C4 , ALL , a]
P3 = P2 | IFFT

In our above program, A1 is the CJ-TN representing the matrix transpose involving all the processors in log *N* time steps, E is the process generating the exponential multipliers, M is the multi-step algorithm. P3 represent the processes containing the result, which is a migrated layer. ALL implies that all the outputs (upto a maximum of *a* which is the data matrix size) are selected. FFT and IFFT are the processes computing the Fast Fourier Transforms of the data. Since the processes are inherently load balanced by the simple structure, the optimization of the code is straightforward (Figure 7).

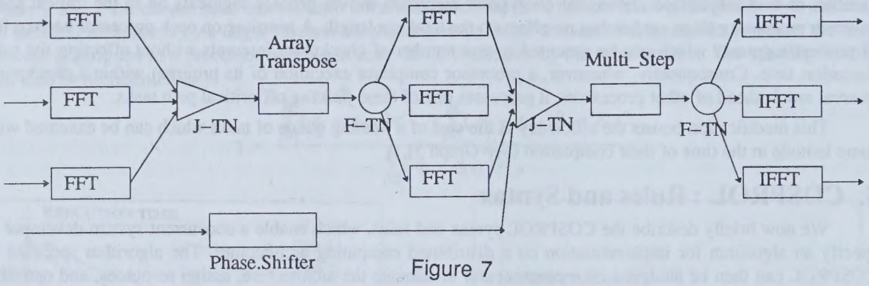


Figure 7

6. Conclusions and Further Work

In this paper, we have discussed distributing computation on a network of multiprogrammed multitasking processors. We have proposed a rigorous model for distributed MIMD computation and analyzed its input output characteristics. Realizable networks were then constructed and algorithms presented which efficiently stabilize such networks and optimize computing resource allocation. A concurrent system programming language COSPROL was proposed which enables the algorithm developer to implement a machine independent description of the computation. Examples of distributed computation paradigms illustrate the flexibility and simplicity of such constructs. Future work will be directed towards implementation of the algorithms proposed here [1,6].

7. Acknowledgement

We thank Prof. J. Walrand for his valuable comments. The authors also thank the review of an earlier version of this paper by Dr. Teresa Meng of Stanford University and by Dr. Francky Cathoor of IMEC, Belgium.

8. References

[1] Madisetti, V.K., D.G. Messerschmitt (1988), **Seismic Migration Algorithms on Parallel Computers,** *Proc. of Third Conference on Hypercube Concurrent Computers and Applications,* Jan 19-20, 1988, JPL-Caltech. To appear IEEE Trans. on ASSP.

[2] Nelson, P., L. Snyder,(1987) **Programming Paradigms for Nonshared Memory Parallel Computers** *CPA, Ed. Jamieson, Gannon and Douglass, MIT Press, 1987.*

[3] Zeman, J. and Moshytz, G.S. (1983), **Systematic Design and Programming of Signal Processors Using Project Management Techniques,** *IEEE Trans. on Acoustics, speech and Signal Processing,* **ASSP-31, December 1983.**

[4] Peterson, J. L. (1977), **Petri Nets,** *Computing Surveys,* **9, September 1977.**

[5] Walrand, J. (1987) *Fundamentals of Queueing Systems,* Prentice-Hall Inc. Englewood Cliffs, 1987.

[5] Athas, W., C. Seitz, (1988), **Multicomputers: Message Passing Concurrent Computers** *IEEE Computer,* August 1988.

[6] Madisetti, V. and J. Walrand, (1988). *Notes on Distributed Discrete Event Simulation.* In preparation.

NEW RESULTS ON EIGENVALUE CONCEPTS FOR LINEAR TIME-VARYING DYNAMICAL SYSTEMS

J. ZHU and C. D. JOHNSON
Electrical and Computer Engineering Department
University of Alabama in Huntsville
Huntsville, AL 35899

ABSTRACT In this paper, a new (time-varying) eigenvalue concept is established for time-varying Vector Linear Dynamical Systems (VLDS) of the form $\dot{x}=A(t)x$, and Scalar LDS (SLDS) of the form $y^{(n)}+\sum_{k=1}^{n}\alpha_k(t)\,y^{(k-1)}$, $y^{(m)}=d^m t/dt^m$. Using this concept, new analytical solution techniques for time-varying VLDS and SLDS are derived. In addition, a necessary and sufficient stability criterion for the general time-varying SLDS is obtained in terms of these new time-varying eigenvalues. Several worked examples are included to demonstrate the power of these new results. The present paper, together with a series of recent papers [8]-[12], constitute a unified "eigen-theory" for both time-invariant and time-varying linear dynamical systems.

1. Introduction

This paper is concerned with extending the concept of *"eigenvalues"* to the class of vector n-dimensional linear time-varying systems

$$\dot{x}=A(t)x, \qquad x(t_0)=x_0, \qquad t\geq t_0, \tag{1.1}$$

and scalar n-th order linear time-varying systems

$$y^{(n)} + \alpha_n(t)y^{(n-1)} + \cdots + \alpha_2(t)\dot{y} + \alpha_1(t)y = 0,$$
$$y^{(k)}(t_0)=y_k, \quad k=1,2,\cdots,n-1, \quad y(t_0)=y_0, \quad t\geq t_0. \tag{1.2}$$

For notational convenience, in the sequel we shall assume $t_0=0$.

It is well-known that the classical concept of eigenvalues λ_i for the time invariant cases of (1.1) and (1.2) (i.e. the n roots λ_i of the associated characteristic equations of (1.1) and (1.2)) does not carry over to the time-varying cases (1.1), (1.2), in general. For instance, one *cannot* reliably predict stability of (1.1) from examination of the "time-varying" roots of $det(A(t)-\lambda(t)I)=0$. It now appears that an effective *"eigen-theory"* for time-varying systems (1.1), (1.2) requires introduction of a generalized concept we have called *Essential D-eigenvalues* (ED-eigenvalues) [9]. This concept can be traced-back to an 1827 paper of Cauchy [2] and an 1879 paper by Floquet [3], but its key role in the solution and stability characterization of (1.1), (1.2) has been established only recently [8], [9], [10]. In this paper we use the ED-eigenvalue concept to develop some important new analytical solution techniques for time-varying linear systems (1.1) and (1.2). Then, we develop a *necessary* and *sufficient* stability criteria for *general time-varying* linear system (1.2).

In order to develop our main results, we first, briefly recall (in Section 2) some essentials of our recent results on ED-eigenvalues and D-similarity transformations. Based on those results, we then introduce, in Section 3, the (new) notion of *Primary ED-eigenvalues*, and the associated (new) notions of *Primary D-characteristic equation, Primary D-spectrum*, and the *S3 Spectral canonical form* for systems (1.1), (1.2). A *canonical Change of Coordinate (CC) transformation matrix* $V(t)$ is also developed that carries the Companion canonical form associated with a scalar system (1.2), or a vector system (1.1), to its corresponding S3 Spectral canonical form. The canonical CC transformation matrix $V(t)$ reduces to the D4 canonical form introduced in [9], and to the classical (time-invariant) Vandermonde matrix form, under appropriate conditions. As part of our main results in this paper, properties of these new entities, and relations between analytical solutions for systems (1.1), (1.2) and these new entities, are uncovered. Finally, using the aforementioned new results, our fundamental results concerning analytical solutions and stability criteria for time-varying linear systems (1.1), (1.2) are established in Section 4.

2. Preliminaries

In this section, we briefly recall some of our recent results on D-similarity transformations and ED-eigenvalues for time-varying systems (1.1) and (1.2).

Let $\mathbb{V}=\mathbb{F}^n$ be the n-dimensional *differential vector space* (D-space) over the *augmented differential field* (D-field) $\mathbb{F}=C^{\infty}_{ae}(\mathbb{R}^+, \mathbb{C}^{\circ})$, which is the set of almost everywhere C^{∞} functions

$f\colon \mathbb{R}^+ \to \mathbb{C}^\circ$, where $\mathbb{R}^+=[0,\infty)$, and $\mathbb{C}^\circ=\mathbb{C}\cup\{0^{-1}\}$ is the augmented field of the complex numbers [8]. The linear time-varying systems (1.1) and (1.2), with $A\in\mathbb{F}^{n\times n}$ and $\alpha_k\in\mathbb{F}$, respectively, can then be studied as *partially linear* [8] operators on $\mathbb{V}$ and $\mathbb{F}$ defined by

$$\mathcal{P}_A\{x\} = [A(t)-I\delta]\{x\}, \tag{2.1}$$

and

$$\mathcal{D}_\alpha\{y\} = [\delta^n+ \sum_{k=1}^{n} \alpha_k(t)\,\delta^{(k-1)}]\{y\}, \tag{2.2}$$

where $\delta=d/dt$. The operators $\mathcal{P}_A$ and $\mathcal{D}_\alpha$ are called, respectively, (n*-dimensional) Vector Linear Differential Equation* (VLDE) *operators*, and (n*-th order) Scalar LDE* (SLDE) *operators.*

Let β and γ be two different ordered basis for $\mathbb{V}$ and let x_β and x_γ be the coordinate vectors for a vector $x\in\mathbb{V}$ relative to the bases β and γ, respectively. Then one can always find an invertible matrix $L\in\mathbb{F}^{n\times n}$ such that $x_\beta=Lx_\gamma$, for all $x\in\mathbb{V}$. The matrix L is known as a *Change of Coordinate* (CC) *transformation matrix*. A *linear* operator $\mathcal{T}$ on $\mathbb{V}$ under CC transformations takes different forms which are said to be *similar*. Now consider a partially linear VLDE operator $\mathcal{P}_A$ on $\mathbb{V}$ with characteristic matrix $A\in\mathbb{F}^{n\times n}$ relative to the ordered basis β. Then

$$\mathcal{P}_A\{x_\beta\}= \mathcal{P}_A\{Lx_\gamma\}= (AL-\dot{L})x_\gamma-L\dot{x}_\gamma=y_\beta, \tag{2.3}$$

which, when expressed with respect to the ordered basis γ, becomes

$$y_\gamma=L^{-1}y_\beta=(L^{-1}AL-L^{-1}\dot{L})x_\gamma-\dot{x}_\gamma=Bx_\gamma-\dot{x}_\gamma=\mathcal{P}_B\{x_\gamma\}, \tag{2.4}$$

where the matrix $B\in\mathbb{F}^{n\times n}$ is given by

$$B=L^{-1}AL-L^{-1}\dot{L}, \tag{2.5}$$

which is the characteristic matrix for the same VLDE operator relative to the basis γ. The transformation from $\mathcal{P}_A$ to $\mathcal{P}_B$ is called a D-similarity transformation if

$$\det L(t)\equiv constant\neq 0. \tag{2.6}$$

In this case, the matrix L is called a *D-similarity transformation matrix*, and the matrices A, B, and their associated VLDE operators $\mathcal{P}_A$, $\mathcal{P}_B$ are said to be *D-similar* on $\mathbb{V}$ [8].

It is well-known that every n-th order time-varying SLDE operator $\mathcal{D}_\alpha$ can be factored into a noncommutative "product" of n first order SLDE operators $\mathcal{D}_{(-\lambda_k)}$ as follows (see [2], [3], and also [4; p121, p133]):

$$\mathcal{D}_\alpha=\mathcal{D}_{(-\lambda_n)}\mathcal{D}_{(-\lambda_{n-1})}\cdots\mathcal{D}_{(-\lambda_1)}=(\delta-\lambda_n)(\delta-\lambda_{n-1})\cdots(\delta-\lambda_1), \tag{2.7}$$

where the scalar functions $\lambda_k(t)$ have been called *roots* [1], *poles* [5], [6], or *ED-eigenvalues* (Essential D-eigenvalues) [8], [9] of the SLDE operator $\mathcal{D}_\alpha$. It is also well-known that the n ED-eigenvalues $\lambda_k(t)$ can be obtained from any fundamental set of n solutions $y_k(t)$ to the equation (1.2) or (2.2) ([3], see also [4; p121]).

It has recently been shown in [9] that every n-dimensional VLDE operator $\mathcal{P}_A$ is related to an n-th order SLDE operator $\mathcal{D}_\alpha$ through an n-dimensional Companion VLDE operaor $\mathcal{P}_C$ that is D-similar to $\mathcal{P}_A$, i.e.,

$$A(t) = L^{-1}(t)\,[C(t)L(t)-\dot{L}(t)] \tag{2.8}$$

for some Companion matrix $C(t)= Comp[-\alpha_1, -\alpha_2, \cdots, -\alpha_n]$ and for some nonsingular matrix $L(t)$ with $\det L(t)= constant \neq 0$. It follows [9] that the ED-eigenvalues of any SLDE operator $\mathcal{D}_\alpha$ that is related to $\mathcal{P}_A$ through (2.8) are also ED-eigenvalues of $\mathcal{P}_A$.

Using the ED-eigenvalues of a time-varying VLDE operator $\mathcal{P}_A$, we have derived, in [9], two (new) *spectral canonical forms* for $\mathcal{P}_A$, called *S1* and *S2 canonical forms*, which resemble the classical *Jordan* and *diagonal spectral canonical forms* in some special cases. An explicit formula was also derived in [9] for constructing a D-similarity transformation (2.8) that transforms $\mathcal{P}_A$ to a Companion canonical form $\mathcal{P}_C$. Moreover, four *canonical D-similarity transformations*, called *D1, D2, D3* and *D4 canonical forms*, respectively, are developed in [9] for transforming a companion VLDE operator $\mathcal{P}_C$ to its S1 and S2 spectral canonical forms. In particular, the D4 canonical form reduces to the classical (constant) *Vandermonde matrix* when all the ED-eigenvalues $\lambda_k(t)\equiv\lambda_k$ are distinct constants, and the D1 canonical form reduces to the well-known (constant) *generalized Vandermonde matrix* when $\lambda_k(t)\equiv\lambda=constant$. An explicit formula for the determinant of a D4 canonical matrix which resembles that for the determinant of a classical (constant) Vandermonde matrix was also given in [9]. Moreover, using the D1 canonical form for $\mathcal{P}_C$, we derived a set of n $(n-k)$-th order nonlinear differential equations [9; Eq. (5.2)]

$$\Delta_k\,(\lambda_1, \lambda_2, \cdots, \lambda_k, \dot{\lambda}_k, \ddot{\lambda}_k, \cdots, \lambda_k^{(n-k)}) = 0; \qquad k=1, 2, \cdots, n, \tag{2.9}$$

whose solutions are the n ED-eigenvalues of $\mathcal{P}_C$. Equations (2.9) are called *a set of D-characteristic equations* for: (i) $\mathcal{P}_C$; (ii) its associated SLDE operator $\mathcal{D}_\alpha$; and (iii) any VLDE operator $\mathcal{P}_A$ that is D-similar to $\mathcal{P}_C$. The results mentioned above leads naturally to the following definition of D-spectrum for SLDE and VLDE operators [9].

DEFINITION 2.1. Let $\mathcal{D}_\alpha$ be an n-th order SLDE operator and let $\Gamma_\alpha=\{\lambda_k(t)\}_{k=1}^n$ be a multi-set (i.e. λ_k need not be distinct) such that $(\delta-\lambda_n)(\delta-\lambda_{n-1})\cdots(\delta-\lambda_1)=\mathcal{D}_\alpha$. Then Γ_α is called a *D-spectrum* for: (i) $\mathcal{D}_\alpha$, (ii) the associated Companion VLDE operator $\mathcal{P}_C$, and (iii) any VLDE operator $\mathcal{P}_A$ that is D-similar to $\mathcal{P}_C$.

REMARKS. In linear operator theory, the definition for a *Spectrum* excludes repeated eigenvalues. However, the multiplicity of repeated eigenvalues of a (time-invariant) linear operator is an important characterization of that operator, and is significant in many applications. Therefore it is desirable to extend the definition of a (time-invariant) spectrum to indicate multiplicity of repeated eigenvalues. Such an extention of a spectrum has been done by many authors. To avoid confusion, in the sequel we shalll call the latter an *extended spectrum.*

3. Primary ED-Eigenvalues

In this section, we first define the (new) notions of *primary ED-eigenvalues, primary D-characteristic equation, and primary D-spectrum,* as extensions of the aforementioned ED-eigenvalue concepts.

DEFINITION 3.1. Let $\mathcal{D}_\alpha$ be an n-th order SLDE operator such that

$$\mathcal{D}_\alpha=\delta^n+\alpha_n(t)\delta^{n-1}+\cdots+\alpha_2(t)\delta+\alpha_1(t)=(\delta-\lambda_n(t))(\delta-\lambda_{n-1}(t))\cdots(\delta-\lambda_1(t)). \tag{3.1}$$

A scalar function $\rho(t)$ having the property that $\lambda_1(t)=\rho(t)$ satisfies (3.1) is called a *Primary ED-eigenvalue* (PED-eigenvalue), for: (i) $\mathcal{D}_\alpha$; (ii) its associated Companion canonical VLDE operator $\mathcal{P}_C$; (iii) any VLDE operator $\mathcal{P}_A$ that is D-similar to $\mathcal{P}_C$.

Since every particular solution $\rho(t)$ of the first D-characteristic equation [9]

$$\Delta_1(\rho,\dot{\rho},\ddot{\rho},\cdots,\overset{(n-1)}{\rho})=0 \tag{3.2}$$

satisfies (3.1), it is natural to introduce the following definition.

DEFINITION 3.2. The first D-characteristic equation (3.2) for an n-th order SLDE operator $\mathcal{D}_\alpha$ is called *the Primary D-characteristic equation* (PD-characteristic equation) for : (i) $\mathcal{D}_\alpha$; (ii) its associated Companion canonical VLDE operator $\mathcal{P}_C$; (iii) any VLDE operator $\mathcal{P}_A$ that is D-similar to $\mathcal{P}_C$.

An analogous extension for the notion of D-spectrum is given next.

DEFINITION 3.3. A set of n PED-eigenvalues $\Upsilon=\{\rho_k(t)\}_{k=1}^n$ for an n-th order SLDE operator $\mathcal{D}_\alpha$ is called a *Primary D-spectrum* (PD-spectrum) for: (i) $\mathcal{D}_\alpha$; (ii) its associated Companion canonical VLDE operator $\mathcal{P}_C$; (iii) any VLDE operator $\mathcal{P}_A$ that is D-similar to $\mathcal{P}_C$, if for all $i,j\leq n$, $\rho_i(t_0)\neq\rho_j(t_0)$, for some $t_0\in\mathbb{R}^+$ that is not a singularity for $\mathcal{D}_\alpha$ (i.e., t_0 is a regular point for all $\alpha_k(t)$, $k=1,2,\cdots,n$).

EXAMPLE 3.1. To clarify the ideas presented above, it is instructive to consider the specific second-order, time-invariant SLDE operator $\mathcal{D}_\alpha=\delta^2+\alpha_2\delta+\alpha_1$, with $\alpha_2=\alpha_1=0$, and examine the corresponding notions of the (conventional) eigenvalues γ_1, γ_2, ED-eigenvalues $\lambda_1(t)$, $\lambda_2(t)$, PED-eigenvalues $\rho_1(t)$, $\rho_2(t)$, the conventional spectrum Λ, the extended (conventional) spectrum Γ°, D-spectrum Γ, PD-spectrum Υ, the (conventional) characteristic equation, D-characteristic equations and PD-characteristic equation. The results are

Eigenvalues:
$$\gamma_1=\gamma_2=0 \tag{3.3}$$

ED-eigenvalues:
$$\lambda_1(t)=\frac{\lambda_{10}}{1+\lambda_{10}t},\quad \lambda_{10}=\lambda_1(0),\qquad \lambda_2(t)=-\lambda_1(t); \tag{3.4}$$

PED-eigenvalues:
$$\rho(t)=\frac{\rho_0}{1+\rho_0 t};\quad \rho_0=\rho(0); \tag{3.5}$$

Spectrum:
$$\Lambda=\{0\}; \tag{3.6}$$

Extended spectrum:
$$\Gamma^\circ=\{0,\ 0\}; \tag{3.7}$$

D-spectrum:
$$\Gamma=\Big\{\frac{\lambda_{10}}{1+\lambda_{10}t},\ \frac{-\lambda_{10}}{1+\lambda_{10}t}\Big\};\quad \lambda_{10}=\lambda_1(0); \tag{3.8}$$

PD-spectrum $$\Upsilon=\left\{\frac{\rho_{10}}{1+\rho_{10}t},\ \ \frac{\rho_{20}}{1+\rho_{20}t}\right\};\ \rho_{10}=\rho_1(0)\neq\rho_2(0)=\rho_{20}; \tag{3.9}$$

Characteristic equation: $$\Delta(\gamma)=\gamma^2=0; \tag{3.10}$$

D-characteristic equations: $$\Delta_1(\lambda_1)=\dot{\lambda}_1+\lambda_1^2=0;\ \ \lambda_1(0)=\lambda_{10}; \tag{3.11a}$$

$$\Delta_2(\lambda_1,\ \lambda_2)=-(\lambda_1+\lambda_2)=0; \tag{3.11b}$$

PD-characteristic equation: $$\Delta_1(\rho)=\dot{\rho}+\rho^2=0;\qquad \rho(0)=\rho_0. \tag{3.12}$$

REMARKS.

1. Note that (3.3) is a particular case of (3.4) with $\lambda_1(0)=0$. Similarly, (3.7) is a special case of (3.8).
2. An n-th order *time-invariant* SLDE operator $\mathfrak{D}_\alpha$ may, in fact, have a set of n *time-varying* ED-eigenvalues. However, such an operator always has also a set of n *time-invariant* ED-eigenvalues and that latter set conincides with the set of conventional eigenvalues of $\mathfrak{D}_\alpha$. In fact, that set of time-invariant ED-eigenvalues is unique, i.e., it is the only set of time-invariant ED-eigenvalues, as has been shown in [11].

The main results of this paper stem from the following important classical result, due to Floquet [3], (see also [4; p121], [1], [6]), which will be stated here as a Lemma using PED-eigenvalue terminology.

LEMMA 3.1 [Floquet]. Let $\mathfrak{D}_\alpha$ be an n-th order SLDE operator and let $\rho\in\mathbb{F}$. Then the following statements are equivalent:

(a) $\rho(t)$ is a PED-eigenvalue for $\mathfrak{D}_\alpha$;
(b) $\mathfrak{D}_\alpha\{y\}=0$, where $y=exp\int\rho(t)dt$;
(c) $\rho(t)=\dot{y}(t)y^{-1}(t)$, where $y(t)$ satisfies $\mathfrak{D}_\alpha\{y\}=0$.

In light of Lemma 3.1, the problem of finding a particular solution to $\mathfrak{D}_\alpha\{y\}=0$ (i.e., system (1.2)) amounts to the equivalent problem of finding a PED-eigenvalue. Our first main result in this paper is an explicit formula for obtaining a set of n PED-eigenvalues for an n-th order SLDE operator $\mathfrak{D}_\alpha$ from a D-spectrum of $\mathfrak{D}_\alpha$. That result is stated as follows.

LEMMA 3.2. Let $\Gamma_\alpha=\{\lambda_k(t)\}_{k=1}^n$ be a D-spectrum for an n-th order SLDE operator $\mathfrak{D}_\alpha$. Then an associated set of n PED-eigenvalues $\{\rho_k(t)\}_{k=1}^n$ can be obtained by the formulaes

$$\rho_1(t)=\lambda_1(t),\qquad \rho_k(t)=\lambda_1(t)\ +\ \dot{q}_k(t)q_k^{-1}(t);\qquad k=2,\ 3,\ \cdots,\ n, \tag{3.13b}$$

where

$$q_k(t)=\int\phi_{21}(t)\int\phi_{32}(t)\int\cdots\int\phi_{k,k-1}(t)\,d^{k-1}t, \tag{3.14}$$

and

$$\phi_{ij}(t)=e^{\int(\lambda_i(t)-\lambda_j(t))dt}. \tag{3.15}$$

Proof. Let

$$y_k(t)=e^{\int\rho_k(t)\,dt}=q_k(t)\,e^{\int\lambda_1(t)\,dt} \tag{3.16}$$

and note that

$$(\delta-\lambda_i(t))\{e^{\int\lambda_i(t)\,dt}\int\phi_{i+1,\,i}(t)\int\cdots\int\phi_{k,\,k-1}(t)\,d^{k-i}t\}$$
$$=e^{\int\lambda_{i+1}(t)}\int\phi_{i+2,\,i+1}(t)\int\cdots\int\phi_{k,\,k-1}(t)\,d^{k-i-1}t\,. \tag{3.17}$$

It is readily verified that $\mathfrak{D}_\alpha\{y_k\}=0$, for all $k\leq n$. By Lemma 3.1, $\rho_k(t)$, $k=1,2,\cdots,n$, are PED-eigenvalues for $\mathfrak{D}_\alpha$. □

REMARKS. The set of PED-eigenvalues $\{\rho_k(t)\}_{k=1}^n$ obtained from *any* D-spectrum $\Gamma_\alpha=\{\lambda_k(t)\}_{k=1}^n$ for $\mathfrak{D}_\alpha$ using (3.13) is, in fact, a PD-spectrum for $\mathfrak{D}_\alpha$. This assertion will be justified following our next main result, Theorem 3.1.

The transformation from a D-spectrum to a PD-spectrum for an n-th order SLDE operator $\mathfrak{D}_\alpha$ defined by (3.13) can also be facilitated by a CC transformation on $\mathbb{V}$. This will be demonstrated in a corollary to Theorem 3.1 below. For this purpose, we need to introduce the following definition.

DEFINITION 3.4. Let $\mathfrak{D}_\alpha$ be an n-th order SLDE operator with a PD-spectrum $\Upsilon_\alpha=\{\rho_k(t)\}_{k=1}^n$. Let $C\in\mathbb{F}^{n\times n}$ be the Companion canonical matrix associated with $\mathfrak{D}_\alpha$. Then the diagonal matrix

$$\Upsilon(t)=diag[\rho_1(t),\ \rho_2(t),\ \cdots,\ \rho_n(t)] \tag{3.18}$$

is called a *Spectral canonical form of the third type*, or simply, an *S3 canonical form*, for $C(t)$. The

VLDE operator $\mathfrak{P}_{\Upsilon}$ is then called a *S3 canonical form* for: (i) $\mathfrak{P}_C$, and (ii) any VLDE opertor $\mathfrak{P}_A$ that is D-similar to $\mathfrak{P}_C$.

Using the D3 canonical form given in [9], we can now develop a canonical CC transformation matrix $V(t)$ that reduces a Companion canonical VLDE operator $\mathfrak{P}_C$, associated with an SLDE operator $\mathfrak{D}_\alpha$, to its S3 canonical form $\mathfrak{P}_{\Upsilon}$. This is our second main result, and can be stated as follows.

THEOREM 3.1. Let $\Gamma_\alpha=\{\lambda_k(t)\}_{k=1}^n$ be a D-spectrum for an n-th order SLDE operator $\mathfrak{D}_\alpha$ and let $\Upsilon_\alpha=\{\rho_k(t)\}_{k=1}^n$ be a PD-spectrum obtained from Γ_α using (3.13). Let $S(t)$ be the D3 canonical form given in [9] associated with Γ_α. Then the S3 canonical form $\Upsilon(t)$ associated with Υ_α can be obtained from the Companion matrix $C(t)$ associated with $\mathfrak{D}_\alpha$ by the CC transformation

$$\Upsilon(t)=V^{-1}(t)[C(t)V(t)-\dot{V}(t)], \tag{3.19}$$

where the matrix $V(t)$ is given by

$$V(t)=S(t)D(t), \tag{3.20a}$$

$$=\begin{bmatrix} 1 & 1 & \cdots & 1 \\ \rho_1 & \rho_2 & \cdots & \rho_n \\ \mathfrak{D}_{\rho_1}\{\rho_1\} & \mathfrak{D}_{\rho_2}\{\rho_2\} & \cdots & \mathfrak{D}_{\rho_n}\{\rho_n\} \\ \vdots & & \ddots & \vdots \\ \mathfrak{D}_{\rho_1}^{n-2}\{\rho_1\} & \cdots & \cdots & \mathfrak{D}_{\rho_n}^{n-2}\{\rho_n\} \end{bmatrix}, \tag{3.20b}$$

where $\mathfrak{D}_{\rho_1}=(\delta+\rho_1)$, and $D(t)=\boldsymbol{diag}[1,\ d_2,\ d_3,\ \cdots,\ d_n]$ with

$$d_k(t)=\phi_{k1}(t)q_k^{-1}(t); \qquad k=2,3,\cdots,n, \tag{3.21}$$

where $\phi_{k1}(t)$ and $q_k(t)$ are given by (3.14) and (3.15).

Proof. Let $\Lambda(t)=\boldsymbol{diag}[\lambda_1(t),\ \lambda_2(t),\ \cdots,\ \lambda_n(t)]$ be the S2 canonical form for C(t) associated with Γ_α. Then it is readily verified that

$$\Upsilon(t)=D^{-1}(t)[\Lambda(t)D(t)-\dot{D}(t)]=\Lambda(t)-D^{-1}(t)\dot{D}(t), \tag{3.22}$$

where $D(t)$ is given by (3.21). Note that

$$\Lambda(t)=S^{-1}(t)[C(t)S(t)-\dot{S}(t)], \tag{3.23}$$

Expression (3.19) now follows immediately. To see that $V(t)$ can be expressed in the canonical form (3.20b), note that a Wronskian matrix $W(t)$ for $\mathfrak{D}_\alpha$ in terms of the S2 canonical form $\Lambda(t)$ and the D3 canonical form $S(t)$ is given by [9; Theorem 9.2]:

$$W(t)=S(t)\,e^{\int\Lambda(t)\,dt}=S(t)\,\boldsymbol{diag}\Big[e^{\int\lambda_1(t)\,dt},\ e^{\int\lambda_2(t)\,dt},\ \cdots,\ e^{\int\lambda_n(t)\,dt}\Big]. \tag{3.24}$$

The same Wronskian matrix $W(t)$ can be represented also by $V(t)$ and the S3 canonical form $\Upsilon(t)$ as follows:

$$\begin{aligned} W(t)&=S(t)D(t)D^{-1}(t)\,e^{\int\Lambda(t)\,dt} \\ &=V(t)\,e^{\int\Upsilon(t)\,dt} \\ &=V(t)\,\boldsymbol{diag}\Big[e^{\int\rho_1(t)\,dt},\ e^{\int\rho_2(t)dt},\ \cdots,\ e^{\int\rho_n(t)\,dt}\Big], \\ &=\begin{bmatrix} y_1(t) & y_2(t) & \cdots & y_n(t) \\ \dot{y}_1(t) & \dot{y}_2(t) & \cdots & \dot{y}_n(t) \\ \vdots & \vdots & \ddots & \vdots \\ y_1^{(n-1)}(t) & y_2^{(n-1)}(t) & \cdots & y_n^{(n-1)}(t) \end{bmatrix}, \end{aligned} \tag{3.25}$$

where

$$y_k(t)=e^{\int\rho_k(t)\,dt}. \tag{3.26}$$

Note that

$$y_k^{(p)}(t)=\mathfrak{D}_{\rho_k}^{p-1}\{\rho_k\}\,e^{\int\rho_k(t)\,dt}, \tag{3.27}$$

with

$$\mathfrak{D}^0_{\rho_k}\{x\}=x, \tag{3.28a}$$

$$\mathfrak{D}^p_{\rho_k}\{x\}=\prod_{i=1}^{p}(\delta+\rho_k)\{x\}; \quad p\geq 1 . \tag{3.28b}$$

The canonical expression for $V(t)$ in (3.20b) is now readily verified. □

Theorem 3.1 has the following three important and useful corollaries. The first corollary justifies the statement made in the Remark following Lemma 3.2.

COROLLARY 3.1. Let $\Upsilon_\alpha=\{\rho_k(t)\}_{k=1}^n$ be a set of n PED-eigenvalues of an n-th order SLDE operator $\mathfrak{D}_\alpha$ obtained from *any* D-spectrum $\Gamma_\alpha=\{\lambda_k(t)\}_{k=1}^n$ for $\mathfrak{D}_\alpha$ using (3.13). Then Υ_α is a PD-spectrum for $\mathfrak{D}_\alpha$.

Proof. It follows from (3.25) that $\{y_k\colon y_k(t)=exp\int\rho_k(t)\,dt\}_{k=1}^n$ is a fundamental set of solutions to $\mathfrak{D}_\alpha\{y\}=0$. Therefore for all $i,j\leq n$, $y_i(t)\neq Cy_j(t)$, where C is a constant. Consequently, $\rho_i(t)\neq\rho_j(t)$, where $\rho_i(t)=\dot{y}_i(t)y_i^{-1}(t)$, and $\rho_j(t)=\dot{y}_j(t)y_j^{-1}(t)$, for all $i,j\leq n$. Thus, Υ_α is a PD-spectrum for $\mathfrak{D}_\alpha$. □

The second corollary to Theorem 3.1 gives a CC transformation that relates an S2 canonical form $\Lambda(t)$ with an S3 canonical form $\Upsilon(t)$ for an n-th order SLDE operator $\mathfrak{D}_\alpha$.

COROLLARY 3.2. Let $\Lambda(t)$ be an S2 canonical form for an n-th order SLDE operator $\mathfrak{D}_\alpha$. Let $D(t)$ be a diagonal CC transformation matrix given by (3.21). Then an S3 canonical form $\Upsilon(t)$ for $\mathfrak{D}_\alpha$ can be obtained by (3.22).

The third corollary to Theorem 3.1 provides an alternative expression for the PD-characteristic equation (3.2).

COROLLARY 3.3. Let $\mathfrak{D}_\alpha$ be an n-th order SLDE operator. Then the PD-characteristic equation (3.2) can be expressed as

$$\Delta_1(\rho)=\mathfrak{D}_\rho^{n-1}\{\rho\}+\alpha_n(t)\mathfrak{D}_\rho^{n-2}\{\rho\}+\ \cdots\ +\alpha_3\mathfrak{D}_\rho\{\rho\}+\alpha_2(t)\rho+\alpha_1(t). \tag{3.29}$$

Proof. Let $\alpha(t)$ be the $(n+1)$-dimension row vector

$$\alpha(t)=[\alpha_1(t),\ \alpha_2(t),\ \cdots,\ \alpha_n(t),\ 1], \tag{3.30}$$

and let $w_{n+1}(t)$ and $v_{n+1}(t)$ be the n-dimension row vectors

$$w_{n+1}(t)=[y_1^{(n)}(t),\ y_2^{(n)}(t),\ \cdots,\ y_n^{(n)}(t)]\ , \tag{3.31}$$

and

$$v_{n+1}(t)=[\mathfrak{D}_{\rho_1}^{n-1}\{\rho_1\},\ \mathfrak{D}_{\rho_2}^{n-1}\{\rho_2\},\ \cdots,\ \mathfrak{D}_{\rho_n}^{n-1}\{\rho_n\}]\ , \tag{3.32}$$

where $\{y_k\colon y_k(t)=exp\int\rho_k(t)\,dt\}_{k=1}^n$ is a fundamental set of solutions to $\mathfrak{D}_\alpha\{y\}=0$. By (3.25),

$$\begin{aligned}\alpha(t)\left[\begin{array}{c}W(t)\\ \hline w_{n+1}(t)\end{array}\right]&=\alpha(t)\left[\begin{array}{c}V(t)\\ \hline v_{n+1}(t)\end{array}\right]e^{\int\Upsilon(t)\,dt}\\ &=[\Delta_1(\rho_1),\ \Delta_1(\rho_2),\ \cdots,\ \Delta_1(\rho_n)]e^{\int\Upsilon(t)\,dt}\\ &=[0,\ 0,\ \cdots,\ 0]\end{aligned} \tag{3.33}$$

where $\Delta_1(\cdot)$ is of the form (3.29). Note that $exp\int\Upsilon(t)\,dt$ is nonsingular, and therefore (3.33) implies

$$\Delta_1(\rho_k)=0; \quad k=1,2,\cdots,n. \tag{3.34}$$

Since $\Delta_1(\cdot)$ in Equation (3.33) satisfies, and is satisfied by, *any* PED-eigenvalues of $\mathfrak{D}_\alpha$, it is a legitimate expression for the PD-characteristic equation for $\mathfrak{D}_\alpha$. □

Our next main result further characterizes the canonical CC transformation matrix $V(t)$.

THEOREM 3.2. Let $\Gamma_\alpha=\{\lambda_k(t)\}_{k=1}^n$ be a D-spectrum for an n-th order SLDE operator $\mathfrak{D}_\alpha$ and let $\Upsilon_\alpha=\{\rho_k(t)\}_{k=1}^n$ be a PD-spectrum obtained from Γ_α using (3.13). Let $V(t)$ be the canonical matrix given by (3.20). Then

$$det\,V(t)\ =\ \prod_{k=1}^{n}\ d_k(t)\neq 0, \tag{3.35}$$

where $d_k(t)$, k=1, 2, ⋯, n, are given by (3.21).

Proof. Since $det\,S(t)\equiv 1$ (see [9]), the first equality in (3.35) follows immediately from (3.20). Moreover, from (3.25) we have

$$det\, V(t) = det\, W(t)\; det\, e^{\int \Upsilon(t)\, dt} \neq 0\ , \tag{3.36}$$

since $y_k(t) = exp \int \rho_k(t)\, dt$ are linearly independent solutions to $\mathfrak{D}_\alpha\{y\}=0$. □

Theorem 3.2 has an important corollary. In order to state that result, we first need the following definition.

DEFINITION 3.5. Let $\Gamma_\alpha = \{\lambda_k(t)\}_{k=1}^n$ be a D-spectrum and let $\Upsilon_\alpha = \{\rho_k(t)\}_{k=1}^n$ be a PD-spectrum for an n-th order SLDE operator $\mathfrak{D}_\alpha$. Then

(a) Γ_α is said to be *simple* if $\lambda_i(t) \neq \lambda_j(t)$, for all $i, j \leq n$;

(b) Γ_α is said to be *homogenetic* if $\dot\lambda_i(t) = \dot\lambda_j(t)$, for all $i, j \leq n$. Similarly, Υ_α is said to be *homogenetic* if $\dot\rho_i(t) = \dot\rho_j(t)$, for all $i, j \leq n$.

REMARKS. By definition, PD-spectrum Υ_α is always *simple* in the sense of definition 3.5-(a).

COROLLARY 3.4. Let $\Gamma_\alpha = \{\lambda_k(t)\}_{k=1}^n$ be a D-spectrum for an n-th order SLDE operator $\mathfrak{D}_\alpha$ and let $\Upsilon_\alpha = \{\rho_k(t)\}_{k=1}^n$ be a PD-spectrum obtained from Γ_α using (3.13). Let $V(t)$ be the canonical matrix given by (3.20). Then

(a) *$det\, V(t) \equiv constant$* if and only if Γ_α is homogenetic, i.e., $\dot\lambda_i(t) = \dot\lambda_j(t)$ for all $i, j \leq n$;

(b) *$det\, V(t) \equiv constant \neq 0$* if and only if Γ_α is homogenetic and simple, i.e., $\dot\lambda_i(t) = \dot\lambda_j(t)$, and $\lambda_i(t) \neq \lambda_j(t)$, for all $i, j \leq n$; or equivalently, if and only if Υ_α is homogenetic, i.e., $\dot\rho_i(t) = \dot\rho_j(t)$ for all $i, j \leq n$.

Moreover, in both cases (a) and (b),

$$det\, V(t) = \prod_{i=2}^{n} \prod_{j=1}^{i-1} (\lambda_i(t) - \lambda_j(t)), \tag{3.37}$$

and in the latter case we also have

$$det\, V(t) = \prod_{i=2}^{n} \prod_{j=1}^{i-1} (\rho_i(t) - \rho_j(t)). \quad \Box \tag{3.38}$$

Since Corollary 3.4 is a special case of Theorem 3.2, the proof is omitted. In fact, Equation (3.37) can be verified by actually evaluating $d_k(t)$ in (3.35) using the given conditions. Equation (3.38) can be justified by invoking the following lemma.

LEMMA 3.3. Let $\Gamma_\alpha = \{\lambda_k(t)\}_{k=1}^n$ be a D-spectrum for an n-th order SLDE operator $\mathfrak{D}_\alpha$. If Γ_α is homogenetic and simple, then it is also a PD-spectrum for $\mathfrak{D}_\alpha$.

Proof. Lemma 3.2 follows immediately from the well-known fact that the Cauchy factorization (3.7) is commutative if and only if $\dot\lambda_i(t) = \dot\lambda_j(t)$, for all $i, j \leq n$. □

REMARKS.

1. In case (b) of Corollary 2, $V(t)$ becomes the D4 canonical form given in [9], where the result (3.37) is also stated and proved as Theorem 7.7.
2. Despite the simplicity of Lemma 3.3, it is fundamentally important because it shows that when the (conventional) *spectrum* for a time-invariant SLDE operator $\mathfrak{D}_\alpha$ is extended to the time-varying case, it splits into two different entities: (i) the *D-spectrum*; and (ii) the *PD-spectrum*. Either one of those spectra characterizes $\mathfrak{D}_\alpha$ completely. In other words, in the time-invariant case, if $\mathfrak{D}_\alpha$ is simple (i.e. no multiple eigenvalues), the (conventional) spectrum for $\mathfrak{D}_\alpha$ is also a *particular* D-spectrum and a particular PD-spectrum for $\mathfrak{D}_\alpha$. It is emphasized here that a time-invariant $\mathfrak{D}_\alpha$ has a unique "time-invariant" (conventional) spectrum; however, it has *infinitely many* "time-varying" D-spectra and PD-spectra, as is illustrated by Example 3.1. A detailed discussion of this latter topic is presented in a forthcoming paper [11].
3. It is interesting to note that the canonical CC transformation matrix $V(t)$ in the form of (3.20), and the PD-characteristic equation $\Delta_1(\cdot)=0$ in the form of (3.29), were also discovered independently by Kamen very recently ([6]; 1988) using a totally different approach. In [6] Kamen proposed to further study the case when the SLDE operator $\mathfrak{D}_\alpha$ has multiple "poles" (PED-eigenvalues). This latter step is actually unnecessary, as has been shown by Theorem 3.1 and its Corollaries. It is also remarked that the canonical matrix $V(t)$ in (3.20) was called a "time-varying" Vandermonde matrix in [6] because it *may* reduce to the classical (time-invariant) Vandermonde matrix V when $\mathfrak{D}_\alpha$ is *time-invariant* and *simple.* However, it appears that designation for $V(t)$ will tend to cause confusion since, even for a *time-invariant, simple,* SLDE operator $\mathfrak{D}_\alpha$, the classical *constant* Vandermonde matrix V is only *one* particular choice for the canonical CC transformation matrix $V(t)$, as we have illustrated in Example 3.1. Moreover, *in the general time-varying case, the conventional eigenvalues for $A(t)$ are totally different from the entities we have called ED-eigenvalues and PED-eigenvalues.*

4. Applications of the PED-Eigenvalue Concept

In this section, the preceeding results are used to develop two major contributions to time-varying linear system theory. First, the PED-eigenvalue concept will be used to develop analytical expressions for the *solutions* of the general time-varying linear systems (1.1) and (1.2), respectively. Then, the PED-eigenvalue concept is used to establish a *necessary and sufficient stability criterion for the general time-varying case of (1.2).* To begin, we first present an analytical expresstion for a fundamental set of solutions to system (1.2).

THEOREM 4.1. Let $\mathfrak{D}_\alpha$ be an n-th order SLDE operator with a PD-spectrum $\Upsilon_\alpha=\{\rho_k(t)\}_{k=1}^n$. Let $\Upsilon(t)$ be the S3 canonical form for Υ_α and let $V(t)$ be given by (3.20). Then a fundamental set of solutions $\{y_k(t)\}_{k=1}^n$ to the time-varying linear system (1.2) is given by

$$[y_1(t),\ y_2(t),\ \cdots,\ y_n(t)]=[\ 1,\ 0,\ 0,\ \cdots,\ 0\ ]\,V(t)e^{\int \Upsilon(t)\,dt}, \tag{4.1}$$

Since Theorem 4.1 is a direct consequence of (3.25), the proof is omitted. The next result gives an analytical expression for a fundamental matrix for the vector time-varying linear system (1.1) using PED-eigenvalues.

THEOREM 4.2. Let $\mathcal{P}_A$ be an n-dimensional VLDE operator and let $\mathcal{P}_C$ be one of its associated Companion canonical forms having an S3 canonical form $\Upsilon(t)=diag[\rho_1(t),\ \rho_2(t),\ \cdots,\ \rho_n(t)]$. Let $V(t)$ be given by (3.20) and let $L(t)$ be a D-similarity transformation matrix carrying $\mathcal{P}_A$ to $\mathcal{P}_C$ (e.g. one that is given by Theorem 5.4 of [9]). Then a fundamental matrix $X_A(t)$ for $\mathcal{P}_A$ is given by

$$X_A(t)\ =\ L(t)\,V(t)e^{\int \Upsilon(t)\,dt}. \tag{4.2}$$

The notion of *relative-mean* introduced in [10] will now be used to derive a new and fundamentally important result in stability theory for linear time-varying systems — namely, we will give *a necessary and sufficient stability criterion for a general scalar n-th order linear time-varying system* (1.2) *based on its PED-eigenvalues.* In order to state that result, we first recall the definition of relative-mean from [10].

DEFINITION 4.1. Let $f:I\to\mathbb{R}$ be Lebesgue integrable on an interval $I=[a,\ b]$. *Let* $\phi:I\to\mathbb{R}$ be monotonically increasing on I. Then the *relative-mean* of $f(t)$ on I, with respect to $\phi(t)$, denoted $rm(f,\ \phi)$, is defined by

$$rm(f,\ \phi)=\frac{1}{\phi(b)-\phi(a)}\int_a^b f(\tau)\ d\tau\ . \tag{4.3}$$

If $I=[a,\ \infty)$ and f is locally Lebesgue integrable on I, then the *relative-mean* of $f(t)$ on I, with respect to ϕ, is defined by

$$rm(f,\ \phi)=\limsup_{b\to\infty}\ \frac{1}{\phi(b)}\int_a^b f(\tau)\ d\tau\ . \tag{4.4}$$

THEOREM 4.3. Let $\mathfrak{D}_\alpha$ be an n-th order time-varying SLDE operator with locally Lebesgue integrable coefficients $\alpha_k(t)$. Let $\Upsilon_\alpha=\{\rho_k(t)\}_{k=1}^n$ be a PD-spectrum for $\mathfrak{D}_\alpha$. Then the null solution of the general time-varying linear system $\mathfrak{D}_\alpha\{y\}=0$ (i.e. system (1.2)) is:

(a) *stable*, in the sense of Liapunov, if and only if there exists a real number M such that $\int_0^t Re\,\rho_k(\tau)d\tau\leq M$ for all $k\leq n$, and for all $t\geq 0$;

(b) *asymptotically stable*, in the sense of Liapunov, if and only if for every $k\leq n$, $Re\,\rho_k(t)$ has a *negative* relative-mean over $\mathbb{R}^+=[0,\infty)$, with respect to some monotonically increasing function $\phi(t)$.

Theorem 4.3 is readily proved by first applying Theorems 3.2 and 3.3 in [10] to each of the fundamental solutions $y_k(t)=exp\int\rho_k(t)\,dt$, $k=1,2,\cdots,n$, and then invoking the well-known fact that the null solution of a linear system (1.2) is (Liapunov) stable if and only if (1.2) has a fundamental set of bounded solutions, and it is (Liapunov) asymptotically stable if and only if (1.2) has a fundamental set of bounded solutions that approach zero as $t\to\infty$. The extension of Theorem 4.3 to the vector linear time-varying systems (1.1) is presented in [12].

We now present two examples to illustrate the applications of the results obtained in this paper.

EXAMPLE 4.1. This example demonstrates how the ED-eigenvalue concept can be used to derive the classical D'Alembert "order-reduction formula" for second-order systems (1.2). Suppose that one particular solution $y_1(t)$ for the general second-order case of (1.2) is known. Then

$$\rho_1(t)=\dot{y}_1(t)y_1^{-1}(t)\ , \tag{4.5}$$

is a corresponding PED-eigenvalue of (1.2). Another PED-eigenvalue $\rho_2(t)$ can be obtained by using (3.13), (3.14) and (3.15) with

$$\lambda_1(t)=\rho_1(t) \ , \qquad \lambda_2(t)=-\alpha_2(t)-\lambda_1(t) \ , \tag{4.6}$$

where (4.6b) is derived from the second D-characteristic equation for (1.2) [9; Eq. (5.2)] which is

$$\Delta(\lambda_1, \lambda_2)=\lambda_1+\lambda_2+\alpha_2(t)=0 \ . \tag{4.7}$$

Now by Theorem 3.1, another linearly independent particular solution $y_2(t)$ to (1.2) is given by

$$y_2(t)=e^{\int\rho_2(t)\,dt}=y_1(t)\int y_1^{-2}(t)\,e^{-\int\alpha_2(t)\,dt}\,dt \ , \tag{4.8}$$

which is recognized as the celebrated D'Alembert order-reduction formula [13; p. 99].

EXAMPLE 4.2. In this example we use Theorem 4.3 to analyze stability of the classical Meissner's equation

$$\ddot{y}+\alpha_1(t)y=0 \ , \tag{4.9a}$$

where $\alpha_1(t)=\alpha_1(t+\pi)$ is a periodic, piecewise constant function defined by:

$$\alpha_1(t)=\begin{cases} a-2q & 0<t\leq\tau \\ a+2q & \tau<t\leq\pi \end{cases} , \tag{4.9b}$$

and where a is the "DC bias", and $2q$ is the half-wave peak magnitude, both a, q are real. Meissner's equation, a special case of Hill's equation, has important applications in mechanical and electrical engineering, and in physics. It is well-known that Meissner's equation is analytically solvable, and necessary and sufficient stability criterion for that equation has been derived using classical Floquet Theory, (see, for example [7]). In this example we will use the PED-eigenvalue theory to develop a necessary and sufficient stability criteries for Meissner's equation.

By Equation (5.2) of [9], the set of D-characteristic equations for (4.9) are given by

$$\dot{\lambda}_1+\lambda_1^2+\alpha_1(t)=0, \qquad \lambda_1+\lambda_2=0 \ , \tag{4.10}$$

where (4.10a) is the PD-characteristic equation for (4.9). To find a PED-eigenvalue for (4.9), write (4.10a) as

$$\dot{\rho}+\rho^2=\begin{cases} u^2 & 0<t\leq\tau \\ v^2 & \tau<t\leq\pi \end{cases} , \tag{4.11}$$

where

$$u=i\sqrt{a-2q} \ , \qquad v=i\sqrt{a+2q} \ . \tag{4.12}$$

Then the general solution os (4.11) over $[0, \pi]$ can be obtained as

$$\rho(t)=\begin{cases} \rho_{(0,\tau]}(t) & 0<t\leq\tau \\ \rho_{(\tau,\pi]}(t) & \tau<t\leq\pi \end{cases} \tag{4.13}$$

where

$$\rho_{(0,\tau]}(t)=\frac{ue^{ut}-C_1ue^{-ut}}{e^{ut}+C_1e^{-ut}}, \qquad \rho_{(t,\pi]}(t)=\frac{ve^{vt}-C_2ve^{-vt}}{e^{vt}+C_2e^{-vt}} \ , \tag{4.14}$$

where C_1, C_2 are constants of integrations to be determined from the initial conditions $\rho_{(0,\tau]}(0)$, and $\rho_{(\tau,\pi]}(\tau)$. Setting

$$\rho_{(0,\tau]}(\tau)=\rho_{(\tau,\pi]}(\tau), \qquad \rho_{(0,\tau]}(0)=\rho_{(\tau,\pi]}(\pi) \ . \tag{4.15}$$

and solving the set of two (quadratic) equations (4.15) to obtain two pairs of C_{1k}, C_{2k}, $k=1,2$, yields two *continuous* and *periodic* solution $\rho_k(t)$, $k=1,2$. Now applying part (b) of Theorem 4.3 to $\rho_k(t)$, and noting that, owing to the periodicity of $\rho_k(t)$, the relative-mean of $\rho_k(t)$ over $(0, \infty]$ can be replaced by the regular mean over one period $(0, \pi]$, we have

$$\frac{1}{\pi}\int_0^\pi \rho_k(t)\,dt=ln\left[\frac{(e^{u\tau}+C_{1k}e^{-\tau})(e^{v\pi}+C_{2k}e^{-v\pi})}{(1+C_{1k})(e^{v\tau}+C_{2k}e^{-v\tau})}\right]=\mu_k; \qquad k=1,2. \tag{4.16}$$

Therefore, in terms of PED-eigenvalues the stability of the null solution of Meissner's equation (4.9) is determined by: (i) unstable if and only if $Re\,\mu_k>0$; and (ii) stable if and only if $Re\,\mu_k\leq 0$, $k=1,2$.

REMARKS. Note that the constants μ_k obtained in (4.16) are actually the Floquet characteristc exponents for (4.9). Note also how the problem of finding a pair of time-varying PED-eigenvalus $\rho_k(t)$ is simplified by choosing a set of particular initial conditions. This solution technique can be generalizedand to the general class of time-varying systems (1.1) and (1.2), and further results on this issue will appear elsewhere.

5. Summary

In this paper we have established the new notions of PED-eigenvalues, PD-spetrum, PD-characteristic equation, for VLDE and SLDE operators. In addition, we have also obtained a canonical change of coordinate transformation matrix that plays an important role in the analysis of time-varying linear systems (1.1), (1.2). Using those results, we have derived: (i) new analytical expressions; (ii) a necessary and sufficient stability criterion, for the general time-varying linear systems (1.1), (1.2), in terms of PED-eigenvalues. These results, together with those in [8], [9], [11], constitute a unified eigen-theory for both time-invariant and time-varying linear dynamical systems (1.1) and (1.2).

REFERENCE

[1] Amitsur, A. S. *Differential Polynomials and Division Algebras,* Annals of Math., Vol. 59, No. 2, 245-278, 1953.

[2] Cauchy, A. *Sur l'analogie des puissances et des differences,* Exercises de Math. 2, 159-209, Paris; Oeuvres d'Augustin Cauchy (Series 2) 7, 198-254, 1827.

[3] Floquet, G. *Annales Scientifiques de l'E cole Normale Superieure,* (2) 8, p.49, 1879.

[4] Ince, E. L. *Ordinary Differential Equations,* Dover Publications, Inc., New York, 1956 (1st ed. 1926).

[5] Kamen, E. W. *Poles and Zeros of a Linear Time-Varying System,* Proc. of Conference on Info. Sci. and Sys., Princeton, NJ, 802-807, 1986.

[6] Kamen, E. W. *The Poles and Zeros of a Linear Time-Varying System,* Linear Alg. and Its Appl., Vol. 98, 263-289, 1988.

[7] Richards, J. A. *Analysis of Periodically Time-Varying Systems*, Springer-Verlag, New York 1983.

[8] Zhu, J. and Johnson, C. D. *New Results on the Reduction of Linear Time-Varying Dynamical Systems,* Proc. IEEE 26th Conference on Decision and Control, Los Angeles, Dec. 1987, To appear in SIAM J. on Control and Optimization.

[9] Zhu, J. and Johnson, C. D. *Unified Cononical Forms for Linear Time-Varying Dynamical Systems Under D-similarity Transformations,* Submitted, 1988.

[10] Zhu, J. and Johnson, C. D. *New Results for the Stability Analysis of Time-Varying Linear Systems; Part I: The Case of Reduced Systems,* Procedings, 1988 American Control Conference, Atlanta, GA, June, 1988.

[11] Zhu, J. and Johnson, C. D. *Invariants of LDE Operators under D-similarity Transformations,* To appear.

[12] Zhu, J. and Johnson, C. D. *New Results for the Stability Analysis of Time-Varying Linear Systems; Part II: The Case of Reducible Systems,* To appear.

[13] Boyce, W. and DiPrima, R. C. *Elementary Differential Equations and Boundary Value Problems,* John Wiley & Sons, New York, 3rd Ed., 1977.

Impulse Responses and Kernel Representations

DAVID BALL and IRWIN W. SANDBERG

Department of Electrical and Computer Engineering
The University of Texas at Austin
Austin, Texas 78712

Abstract

Results are given that establish, for the first time, necessary and sufficient conditions for the existence of impulse responses and kernel representations for linear not-necessarily-time-invariant systems described by input-output operator equations.

These results concern systems whose inputs and outputs are real-valued functions on the real line R, the half line $[0, \infty)$, or R^m. They deal with causal as well as noncausal maps, and considerably extend related previous results which concern causal maps defined on functions on the half line.

1 Introduction

In a recent study [1] concerned in part with the representation of causal linear operators H mapping locally integrable functions on $[0, \infty)$ to functions on $[0, \infty)$, sufficient conditions are given under which H has the representation

$$(Hu)(t) = \int_0^t h(t, \tau) u(\tau) d\tau, \qquad t > 0 \tag{1}$$

for each input u, where the kernel h, the impulse response of the system, is an ordinary (as opposed to generalized) function, and the integral is a Lebesgue integral. If H is time-invariant, (1) becomes

$$(Hu)(t) = \int_0^t g(t - \tau) u(\tau) d\tau, \qquad t > 0, \tag{2}$$

where, given $t > 0$, $g(\tau) = h(t, t - \tau)$ for almost all τ in $[0, t]$. Necessary and sufficient conditions are given in [1] under which g in (2) is continuous.

The relation (1) was obtained as a consequence of a representation result in [1] for causal nonlinear maps defined on functions on the half line. In this paper, we take a different tack, and obtain (1) and related results using linear operator theory and integration theory. By slightly generalizing the conditions in [1] under which (1) holds, we obtain necessary and sufficient conditions for the existence of representations of the form (1). Our approach leads also to criteria for the existence of impulse responses and kernel representations for systems that need not be causal, and for systems whose inputs and outputs are functions on the whole real line R or R^m.

Our results are given in Section 2, which begins with some preliminaries. Section 2.2 contains Theorems 1–3, which provide necessary and sufficient conditions for the existence of impulse responses and kernel representations for maps of functions defined on R, $[0, \infty)$ or R^m, respectively. For example, Theorem 1 provides conditions, which are essentially continuity conditions, under which a map H defined on a space U of locally integrable functions $u : R \rightarrow R$ has the representation

$$(Hu)(t) = \lim_{n \rightarrow \infty} \int_{-n}^{n} h(t, \tau) u(\tau) d\tau, \qquad t \in R, \quad u \in U, \tag{3}$$

and conditions under which (3) reduces to

$$(Hu)(t) = \int_{-\infty}^{\infty} h(t, \tau) u(\tau) d\tau, \qquad t \in R \tag{4}$$

for certain inputs $u \in U$. Section 2.2 also contains one of the highlights of the paper, a corollary to Theorem 1 which provides necessary and sufficient conditions under which a linear map H defined on a space of bounded inputs u has the representation (4). Theorems 1–3 are obtained using the same general approach adapted to specific cases. In this version of the paper, we present a sketch of the proof of Theorem 1 and comments on changes in details of the proof needed to obtain Theorems 2 and 3.

Notes on Theorems 1–3 are presented in Section 2.3. These include an example of a case in which (3) does not reduce to (4).

Results on time-invariant (or, in the R^m case, space-invariant) maps appear in Section 2.4. Theorem 4 gives necessary and sufficient conditions on a linear map H under which H has the representation

$$(Hu)(t) = \lim_{n \to \infty} \int_{-n}^{n} g(t-\tau)u(\tau)d\tau, \qquad t \in R \tag{5}$$

for $u \in U$, and additional conditions under which (5) reduces to

$$(Hu)(t) = \int_{-\infty}^{\infty} g(t-\tau)u(\tau)d\tau, \qquad t \in R. \tag{6}$$

Section 2.4 also contains Theorem 5, which provides a complete solution to the problem of obtaining conditions under which g in (5) or (6) is continuous.

In order to present as many results as possible in this version of the paper, we have omitted all proofs.

2 Representation Results

2.1 Preliminaries

We first consider some basically standard definitions and notation, in which R and R^m denote the set of real numbers and the set of real m-vectors, respectively.

Let $(X, \mathcal{M}, \mu)$ be a measure space. The set of functions $f : X \to R$ which are (Lebesgue) integrable with respect to μ is denoted by $\mathcal{L}^1(\mu)$. Given $f \in \mathcal{L}^1(\mu)$, we write the integral of f over X as $\int f d\mu$ or, if f is a function of several variables and, say, $f(a,\cdot) \in \mathcal{L}^1(\mu)$, we use $\int f(a,\cdot)d\mu$ to denote the integral of $f(a,\cdot)$ over X. The integral of f over a set $A \in \mathcal{M}$ is written as $\int_A f\, d\mu$ or $\int f \cdot \chi_A d\mu$, where χ_A denotes the characteristic function of A. The set of functions $f : X \to R$ that are locally integrable, i.e., satisfy $f \cdot \chi_A \in \mathcal{L}^1(\mu)$ for each set $A \in \mathcal{M}$ satisfying $\mu(A) < \infty$, is denoted by $\mathcal{L}^1_{\text{loc}}(\mu)$. The set of functions $f : X \to R$ which are μ-measurable and bounded except on a set of μ-measure zero is denoted by $\mathcal{L}^\infty(\mu)$.

We denote Lebesgue measure on R, $[0,\infty)$ or R^m by λ, and write $\mathcal{L}^1(R)$, $\mathcal{L}^1(0,\infty)$ or $\mathcal{L}^1(R^m)$ instead of $\mathcal{L}^1(\lambda)$ (and similarly for $\mathcal{L}^1_{\text{loc}}(\lambda)$ and $\mathcal{L}^\infty(\lambda)$). Given f in $\mathcal{L}^1(R)$, $\mathcal{L}^1(0,\infty)$ or $\mathcal{L}^1(R^m)$, we write, e.g., $\int_{R^m} f(\tau)d\tau$ instead of $\int f\, d\lambda$ and $\int_a^b f(\tau)d\tau$ instead of $\int_{(a,b)} f\, d\lambda$. Given f in $\mathcal{L}^1(R^m)$, we often write, e.g., $\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\tau_1,\tau_2)\, d\tau_1 d\tau_2$ instead of $\int_{R^2} f(\tau)d\tau$, where $\tau = (\tau_1,\tau_2)$ (of course this is permissable by Fubini's Theorem). The standard $\mathcal{L}^1(R)$ ($\mathcal{L}^1(0,\infty)$, $\mathcal{L}^1(R^m)$) and $\mathcal{L}^\infty(R)$ ($\mathcal{L}^\infty(0,\infty)$, $\mathcal{L}^\infty(R^m)$) norms are denoted by $\|\cdot\|$ and $\|\cdot\|_\infty$ respectively. Whenever we say that a property holds almost everywhere, or that a function or set is measurable, we are referring to Lebesgue measure unless otherwise indicated.

Given a set X, we define the unit constant function $1 : X \to R$ by $1(t) = 1$, $t \in X$.

Finally, whenever we say that a limit exists, we mean that it exists as a real number.

2.2 Representations for Time-Varying Maps

Theorems 1–3 of this section provide necessary and sufficient conditions for the existence of impulse responses and kernel representations for linear systems whose inputs and outputs are real-valued functions defined on R, $[0,\infty)$ or R^m, respectively. As is mentioned in the Introduction, each of these theorems can be proved using the same general approach. We include after Theorem 1 an outline of the steps in obtaining a proof of any of these theorems, and we state Theorems 2 and 3 with only a few comments concerning changes in the hypotheses and conclusions of Theorem 1.

2.2.1 Theorems 1 and 2

We use the following notation in connection with maps of functions defined on the whole line R or the half line $[0,\infty)$.

Let $M(\tilde{M})$ denote the set of functions $f : R \to R$ ($f : [0,\infty) \to R$). Given τ in R, we define the delay or shift operator $T_\tau : M \to M$ by

$$(T_\tau f)(t) = f(t-\tau), \qquad t \in R, \quad f \in M.$$

The delay operator $T_\tau : \tilde{M} \to \tilde{M}$, $\tau \geq 0$, is defined by

$$(T_\tau f)(t) = \begin{cases} 0, & t < \tau \\ f(t-\tau), & t \geq \tau \end{cases}, \quad f \in \tilde{M}.$$

We use $W(\tilde{W})$ to denote the set of functions of bounded support in $M(\tilde{M})$ that are piecewise-constant and have a finite number of discontinuities. Let $U(\tilde{U})$ denote any linear space over R ($[0,\infty)$) satisfying

(a) $W \subseteq U \subseteq \mathcal{L}^1_{\text{loc}}(R)$ $\left(\tilde{W} \subseteq \tilde{U} \subseteq \mathcal{L}^1_{\text{loc}}(0,\infty)\right)$,

(b) $U(\tilde{U})$ is closed under T_τ for each τ in R ($[0,\infty)$) and is closed under multiplication with the characteristic functions of intervals in R ($[0,\infty)$) (e.g., $u \in U$ implies $u \cdot \chi_{[a,b]} \in U$ for all a and b in R with $a \leq b$).

For example, U can be taken to be W, or, at the other extreme, we can take $U = \mathcal{L}^1_{\text{loc}}(R)$.

Given $t \in R$, let $L(t)$ be the set of linear maps $H : U \to M$ such that, given $u \in U$, $\lim_{n\to\infty} \left(H\chi_{[-n,n]}u\right)(t)$ exists and equals $(Hu)(t)$. Let $\tilde{L}(t)$, $t \geq 0$, denote the set of linear maps $H : \tilde{U} \to \tilde{M}$ such that $\lim_{n\to\infty} \left(H\chi_{[0,n]}u\right)(t)$ exists and equals $(Hu)(t)$ for each $u \in \tilde{U}$. The (very reasonable) hypothesis that an input-output map belongs to $L(t)$ or $\tilde{L}(t)$ (or the set $L_m(t)$ defined in Section 2.2.2) plays a central role in our results.

We say that a map $H : U \to M$ is causal if $(Hu)(t) = \left(H\chi_{(-\infty,\alpha]}u\right)(t)$, $u \in U, t \leq \alpha$, for each $\alpha \in R$. A map $H : \tilde{U} \to \tilde{M}$ is causal if for each $u \in \tilde{U}$ and $\alpha \geq 0$, $(Hu)(t) = \left(H\chi_{[0,\alpha]}u\right)(t)$, $t \leq \alpha$. We say that a map $H : U \to M$ $\left(H : \tilde{U} \to \tilde{M}\right)$ is time-invariant if $T_\alpha H = HT_\alpha$ for each α in R ($[0,\infty)$).

Given $\sigma > 0$, we define w_σ in W or $\tilde{W}$ by

$$w_\sigma(t) = \begin{cases} \dfrac{1}{\sigma}, & t \in [0,\sigma) \\ 0, & \text{otherwise} \end{cases}.$$

In either space, w_σ approaches the unit impulse at the origin (in the usual intuitive sense).

In Theorem 1, we are concerned with subsets $B(w,t)$ of $L(t)$ defined as follows: for each $t \in R$ and measurable $w \in M$, $B(w,t) = \left\{H \in L(t) : |(Hu)(t)| \leq \|wu\| \text{ for all } u \in U \text{ satisfying } wu \in \mathcal{L}^1(R)\right\}$. We now present Theorem 1.

Theorem 1 Let $H : U \to M$ be linear, and let $t \in R$. Then

1) the condition that $H \in B(w,t)$ for some $w \in \mathcal{L}^\infty(R)$ is necessary and sufficient for the following to hold:

   (a) $h(t,\tau) \triangleq \lim_{\sigma\to 0} (HT_\tau w_\sigma)(t)$ exists for almost all τ in R,

   (b) the function $h(t,\cdot)$ belongs to $\mathcal{L}^\infty(R)$,

   and

   (c) $\lim_{n\to\infty} \int_{-n}^{n} h(t,\tau)u(\tau)d\tau$ exists for each $u \in U$, and we have

$$(Hu)(t) = \lim_{n\to\infty} \int_{-n}^{n} h(t,\tau)u(\tau)d\tau, \qquad u \in U, \tag{7}$$

and

2) (a)–(c) above hold with $h(t,\cdot)$ belonging to $\mathcal{L}^1(R)$ if and only if $H \in B(w,t)$ for some $w \in \mathcal{L}^1(R) \cap \mathcal{L}^\infty(R)$.

Notes

1. If $(Hu)(t)$ can be expressed as in (7) with $h(t,\cdot) \in \mathcal{L}^\infty(R)$, and if $u \in \mathcal{L}^1(R)$, then

$$(Hu)(t) = \int_{-\infty}^{\infty} h(t,\tau)u(\tau)d\tau. \tag{8}$$

Similarly, if $(Hu)(t)$ is given by (7) with $h(t,\cdot) \in \mathcal{L}^1(R)$ and $u \in \mathcal{L}^\infty(R)$, then (8) holds.

2. The following important corollary to Theorem 1 provides representations for a class of linear maps on $\mathcal{L}^\infty(R)$ (similar corollaries to Theorems 2 and 3 are easily obtained, and these are given in the Appendix).

   Corollary 1 Let $H : \mathcal{L}^\infty(R) \to M$ be linear, and let $t \in R$. Then the condition that there exists $w \in \mathcal{L}^1(R) \cap \mathcal{L}^\infty(R)$ such that $|(Hu)(t)| \leq \|wu\|$ for all $u \in \mathcal{L}^\infty(R)$ is necessary and sufficient for the following to hold:

   (a) $h(t,\tau) \triangleq \lim_{\sigma\to 0} (HT_\tau w_\sigma)(t)$ exists for almost all τ in R,
   (b) the function $h(t,\cdot)$ belongs to $\mathcal{L}^1(R) \cap \mathcal{L}^\infty(R)$, and
   (c) we have

   $$(Hu)(t) = \int_{-\infty}^{\infty} h(t,\tau)u(\tau)d\tau, \quad u \in \mathcal{L}^\infty(R).$$

   By modifying the proof of Theorem 1, one may obtain a similar result in which w is integrable but not necessarily bounded (in this case, one gets $h(t,\cdot) \in \mathcal{L}^1(R)$, but not $h(t,\cdot) \in \mathcal{L}^\infty(R)$). Note that, adopting the viewpoint that $(H(\cdot))(t)$ is a linear functional, Corollary 1 provides Riesz-type representations for a class of bounded linear functionals on $\mathcal{L}^\infty(R)$.

3. Theorem 1 provides a representation for $(Hu)(t)$ for arbitrary t and u; it therefore yields a representation for H. Specifically, it is clear from the theorem that if there is a function $w : R \times R \to R$ such that $w(t,\cdot) \in \mathcal{L}^\infty(R)$ and $H \in B(w(t,\cdot),t)$ for each $t \in R$, then

   $$(Hu)(t) = \lim_{n\to\infty} \int_{-n}^{n} h(t,\tau)u(\tau)d\tau, \quad t \in R, \quad u \in U,$$

   where $h(t,\cdot)$ is defined as in (a) for each $t \in R$.

4. The condition of Theorem 1 that $H \in B(w,t)$ for some $w \in \mathcal{L}^\infty(R)$ is essentially a continuity condition on H. It requires that H be in $L(t)$, i.e., that the response $(Hu)(t)$ be the limit of a sequence of responses to truncations of u, and that $(H(\cdot))(t)$ be a bounded (and therefore continuous) linear functional on the space of functions $u \in U$ satisfying $wu \in \mathcal{L}^1(R)$ (see (β) in the proof of Theorem 1, below).

5. It will become clear that similar comments apply to Theorems 2 and 3 and to Theorem 4 in Section 2.4. Further notes on Theorems 1–3 appear in Section 2.3.

Proof of Theorem 1 The steps in the proof are as follows:

1) *Necessity.* Observe that $H \in B(h(t,\cdot),t)$.
   Sufficiency. (α) Define a positive measure φ on the Lebesgue measurable sets A in R by

   $$\varphi(A) = \int_A |w(\tau)|\,d\tau.$$

   Show that $\mathcal{L}^1(\varphi)$ is the set of functions f satisfying $wf \in \mathcal{L}^1(R)$.
   (β) Notice that, since $H \in B(w,t)$, $(H(\cdot))(t)$ is a bounded linear functional on the subspace $U \cap \mathcal{L}^1(\varphi)$ of $\mathcal{L}^1(\varphi)$. Use the Hahn-Banach Extension Theorem to extend $(H(\cdot))(t)$ to a bounded linear functional on $\mathcal{L}^1(\varphi)$.
   (γ) Use the Riesz Representation Theorem and a fundamental result on the differentiation of an integral to obtain (a) and (b) and the representation

   $$(Hu)(t) = \int_{-\infty}^{\infty} h(t,\tau)u(\tau)d\tau$$

   for each $u \in M$ satisfying $wu \in \mathcal{L}^1(R)$.
   (δ) Notice that (c) follows from (γ) and the fact that $H \in L(t)$.

2) Use an argument similar to that in (γ) above to show that $h(t,\cdot) \in \mathcal{L}^1(R)$ when $w \in \mathcal{L}^1(R) \cap \mathcal{L}^\infty(R)$. For the converse, notice that since $H \in B(h(t,\cdot),t)$ when (a)–(c) hold, we have that $h(t,\cdot) \in \mathcal{L}^1(R)$ implies $H \in B(w,t)$ for some $w \in \mathcal{L}^1(R) \cap \mathcal{L}^\infty(R)$.

Now consider systems whose inputs and outputs are defined on the half line $[0,\infty)$. An examination of the proof of Theorem 1 reveals that similar results hold for maps $H : \tilde{U} \rightarrow \tilde{M}$ in $\tilde{L}(t)$ for $t \geq 0$. Given a measurable $w \in \tilde{M}$ and $t \geq 0$, define $\tilde{B}(w,t) = \{H \in \tilde{L}(t) : |(Hu)(t)| \leq \|wu\| \text{ for each } u \in \tilde{U} \text{ satisfying } wu \in \mathcal{L}^1(0,\infty)\}$.

Theorem 2 Let $H : \tilde{U} \rightarrow \tilde{M}$ be linear, and let $t \geq 0$. Then the condition that $H \in \tilde{B}(w,t)$ for some $w \in \mathcal{L}^\infty(0,\infty)$ is necessary and sufficient for the following to hold:

(a) $h(t,\tau) \triangleq \lim_{\sigma\rightarrow 0} (HT_\tau w_\sigma)(t)$ exists for almost all τ in $[0,\infty)$,

(b) the function $h(t,\cdot)$ belongs to $\mathcal{L}^\infty(0,\infty)$,

and

(c) $\lim_{n\rightarrow\infty} \int_0^n h(t,\tau)u(\tau)d\tau$ exists for each $u \in \tilde{U}$, and we have

$$(Hu)(t) = \lim_{n\rightarrow\infty} \int_0^n h(t,\tau)u(\tau)d\tau, \qquad u \in \tilde{U}.$$

Furthermore, (a)–(c) above hold with $h(t,\cdot)$ belonging to $\mathcal{L}^1(0,\infty)$ if and only if $H \in \tilde{B}(w,t)$ for some $w \in \mathcal{L}^1(0,\infty) \cap \mathcal{L}^\infty(0,\infty)$.

2.2.2 Theorem 3

We now address the matter of extending Theorem 1 to cover the case of maps of functions defined on R^m. We will use the following notation.

Let M_m denote the set of functions $f : R^m \rightarrow R$. Given $\tau \in R^m$, we define the shift operator $T_\tau : M_m \rightarrow M_m$ by

$$(T_\tau f)(t) = f(t-\tau), \quad t \in R^m, \quad f \in M_m.$$

Define W_m as the set of functions of bounded support in M_m that are piecewise-constant and have a finite number of discontinuities. Let U_m denote any linear space over R^m satisfying

(a) $W_m \subseteq U_m \subseteq \mathcal{L}^1_{\text{loc}}(R^m)$,

(b) U_m is closed under T_τ for each $\tau \in R^m$ and is closed under multiplication with the characteristic functions of intervals in R^m (an interval in R^m is the direct product of intervals in R).

Given $t \in R^m$, let $L_m(t)$ be the set of linear maps $H : U_m \rightarrow M_m$ such that, given $u \in U_m$, $\lim_{n\rightarrow\infty} \left(H\chi_{[-n,n]^m} u\right)(t)$ exists and equals $(Hu)(t)$.

Given $\sigma > 0$, we define a function $w_\sigma \in W_m$ by

$$w_\sigma(t) = \begin{cases} \dfrac{1}{\sigma^m}, & t \in [0,\sigma)^m \\ 0, & \text{otherwise} \end{cases}.$$

As $\sigma \rightarrow 0$, w_σ approaches the unit m-dimensional impulse at the origin (in the usual intuitive sense).

As is the case with Theorem 2, conclusions similar to those of Theorem 1 hold for $H \in L_m(t)$, $t \in R^m$. Given a measurable $w \in M_m$ and $t \in R^m$, define $B_m(w,t) = \{H \in L_m(t) : |(Hu)(t)| \leq \|wu\| \text{ for each } u \in U_m \text{ satisfying } wu \in \mathcal{L}^1(R^m)\}$.

Theorem 3 Let $H : U_m \rightarrow M_m$ be linear, and let $t \in R^m$. Then the condition that $H \in B_m(w,t)$ for some $w \in \mathcal{L}^\infty(R^m)$ is necessary and sufficient for the following to hold:

(a) $h(t,\tau) \triangleq \lim_{\sigma\rightarrow 0} (HT_\tau w_\sigma)(t)$ exists for almost all τ in R^m,

(b) the function $h(t,\cdot)$ belongs to $\mathcal{L}^\infty(R^m)$,

and

(c) $\lim_{n\rightarrow\infty} \int_{[-n,n]^m} h(t,\tau)u(\tau)d\tau$ exists for each $u \in U_m$, and we have

$$(Hu)(t) = \lim_{n\rightarrow\infty} \int_{[-n,n]^m} h(t,\tau)u(\tau)d\tau, \qquad u \in U_m.$$

Furthermore, (a)–(c) above hold with $h(t,\cdot)$ belonging to $\mathcal{L}^1(R^m)$ if and only if $H \in B_m(w,t)$ for some $w \in \mathcal{L}^1(R^m) \cap \mathcal{L}^\infty(R^m)$.

2.3 Notes on Theorems 1–3

Theorems 1–3 are related to Theorem 2 of [1], which is obtained as an application of a representation result in [1] for nonlinear causal maps of functions defined on the half line $[0, \infty)$. Our Theorems 1 and 3 address the doubly-infinite interval and R^m cases, which are not considered in [1]. In addition, we provide necessary and sufficient conditions for the existence of representations, while only sufficient conditions are given in [1]. Our Theorem 2 is more general than Theorem 2 of [1]. The latter is in fact a consequence of the following corollary to Theorem 2, in which $\tilde{B}^c(k,t)$ for $t \geq 0$ and $k \geq 0$ denotes $\left\{H : \tilde{U} \to \tilde{M} : H \text{ is causal and linear, and } |(Hu)(t)| \leq k \left\|\chi_{[0,t]}u\right\| \text{ for each } u \in \tilde{U}\right\}$ (notice that $\tilde{B}^c(k,t) \subseteq \tilde{L}(t)$).

Corollary 2 The condition that $H \in \tilde{B}^c(k,t)$ for some $k \geq 0$ for each $t \geq 0$ is necessary and sufficient for the following to hold:

(a) $h(t,\tau) \triangleq \lim_{\sigma \to 0} (HT_\tau w_\sigma)(t)$ exists for almost all τ in $[0,t]$ for each $t \geq 0$,

(b) the function $h(t,\cdot)$ belongs to $\mathcal{L}^\infty(0,t)$ for each $t \geq 0$,

and

(c) $(Hu)(t) = \int_0^t h(t,\tau)u(\tau)d\tau, \quad t \geq 0, \quad u \in \tilde{U}$.

Much of Theorems 1–3 may be obtained as consequences of the representation results for nonlinear maps given in [2]. The approach used here yields stronger results and has the advantage of directness.

As we mentioned in Section 2.2.1, the requirement that $H \in B(w,t)$ for some $w \in \mathcal{L}^\infty(R)$ is essentially a continuity condition on the map $H : U \to M$. The following result makes this point more explicitly.

Lemma 1 Let $t \in R$, and let $H \in L(t)$. Then in order for there to exist some $w \in \mathcal{L}^\infty(R)$ such that $H \in B(w,t)$, it is necessary and sufficient that there is a function $w' \in \mathcal{L}^\infty(R)$ such that $(Hu_n)(t) \to 0$ as $n \to \infty$ for any sequence $\{u_n\}$ in U satisfying $w'u_n \in \mathcal{L}^1(R)$ for each n and $\|w'u_n\| \to 0$ as $n \to \infty$.

The function $h(t,\cdot)$ defined in Theorems 1–3 may be defined in several equivalent ways. For example, we may define $h(t,\cdot)$ in (a) of Theorem 1 by

$$h(t,\tau) = \lim_{\sigma \to 0} \left(HT_\tau \left(\frac{1}{\sigma}\chi_{(\tau-\sigma,\tau]}\right)\right)(t)$$

for almost all τ in R, because the set $(\tau - \sigma, \tau]$ shrinks "nicely" [3, pp.140–141] to τ as $\sigma \to 0$ (recall that, in this context, $w_\sigma = \frac{1}{\sigma}\chi_{[\tau,\tau+\sigma)}$). Similarly, in R^2 the set $[\tau_1, \tau_1 + \sigma) \times [\tau_2, \tau_2 + \frac{1}{4}\sigma)$ shrinks nicely to $\tau = (\tau_1, \tau_2)$ as $\sigma \to 0$, so we may define $h(t,\cdot)$ in (a) of Theorem 3 by

$$h(t,\tau) = \lim_{\sigma \to 0} \left(HT_\tau \left(\frac{4}{\sigma^2}\chi_{[\tau_1,\tau_1+\sigma)\times[\tau_2,\tau_2+\frac{1}{4}\sigma)}\right)\right)(t)$$

for almost all τ in R^2. However, the set $[\tau_1, \tau_1 + \sigma) \times [\tau_2, \tau_2 + \sigma^2)$ does *not* shrink nicely to τ as $\sigma \to 0$, so we may not define $h(t,\cdot)$ in terms of $\left(HT_\tau \left(\frac{1}{\sigma^3}\chi_{[\tau_1,\tau_1+\sigma)\times[\tau_2,\tau_2+\sigma^2)}\right)\right)(t)$.

As was noted in Section 2.2.1, if the conditions of Theorem 1 are met for a linear map $H : U \to M$ and some $t \in R$, and if $h(t,\cdot)$ defined in (a) of Theorem 1 belongs to $\mathcal{L}^1(R)$, then we have (8) for each $u \in U \cap (\mathcal{L}^1(R) \cup \mathcal{L}^\infty(R))$, i.e., for each (essentially) bounded or integrable u in U. In this case, we have a representation for $(Hu)(t)$ for $u \in U$ of the form (7) but not of the form (8) only if $u \in U$ is neither bounded nor integrable. Such inputs are usually of limited interest.

However, if $h(t,\cdot)$ is not in $\mathcal{L}^1(R)$, then we have (8) only for those inputs $u \in U$ which are integrable, and we must represent $(Hu)(t)$ as a limit of the form (7) for some interesting inputs, such as the unit constant function 1. As an example, let U denote the space of piecewise-constant functions on R (i.e., functions which take on finitely many values and have only a finite number of discontinuities), and define a map $H : U \to M$ by

$$(Hu)(t) = \lim_{n \to \infty} \int_{-n}^{n} \frac{\sin(t-\tau)}{t-\tau} u(\tau)d\tau, \quad t \in R, \quad u \in U. \tag{9}$$

It is clear that the conditions of Theorem 1 are satisfied for each $t \in R$, and that $h(t,\cdot)$, $t \in R$, is defined by $h(t,\tau) = \frac{\sin(t-\tau)}{t-\tau}$ for almost all τ in R. Thus, given $t \in R$, $h(t,\cdot)$ is not in $\mathcal{L}^1(R)$ and, as is clear from (9), we have a representation of the form (8) for $(Hu)(t)$ only for those inputs $u \in U$ which have bounded support. Given $t \in R$, $(H1)(t)$ must be expressed as a limit of the form (7); in fact,

$$(H1)(t) = \lim_{n\to\infty} \int_{-n}^{n} h(t,\tau)d\tau = \lim_{n\to\infty} \int_{-n}^{n} \frac{\sin(t-\tau)}{t-\tau} d\tau, \qquad t \in R,$$

so since $h(t,\cdot)$ is not in $\mathcal{L}^1(R)$, we cannot reduce the limit to the form (8).

2.4 Representations for Time-Invariant Maps

In this section, we consider linear maps that are time-invariant. The results given concern maps of functions defined on the whole line R. Similar results for the half line and R^m cases are easily obtained.

The theorems in this section are related to Theorems 3 and 4 of [1], which concern causal linear maps of functions defined on the half line $[0,\infty)$.

Theorem 4 is our main representation result for linear time-invariant maps. We will refer to the following hypothesis in our statement of the theorem.

A.1. Given t and τ in R and $\sigma > 0$, one has $(HT_\tau v_\sigma)(t) = (Hv_\sigma)(t-\tau)$, where $v_\sigma \in W$ is defined by $v_\sigma = \chi_{[0,\sigma)}$ (notice that $v_\sigma = \sigma w_\sigma$).

We now present Theorem 4.

Theorem 4 Let $H : U \to M$ be linear, and suppose that A.1 holds. Then

1) the condition that $H \in B(w,t)$ for some $w \in \mathcal{L}^\infty(R)$ for each $t \in R$ is necessary and sufficient for the following to hold:

   (a) $g(\tau) \triangleq \lim_{\sigma\to 0}(Hw_\sigma)(\tau)$ exists for almost all τ in R,

   (b) the function g belongs to $\mathcal{L}^\infty(R)$,

   and

   (c) $\lim_{n\to\infty}\int_{-n}^{n} g(t-\tau)u(\tau)d\tau$ exists for each $u \in U$ and $t \in R$, and we have

$$(Hu)(t) = \lim_{n\to\infty}\int_{-n}^{n} g(t-\tau)u(\tau)d\tau, \qquad t \in R, \quad u \in U,$$

   and

2) (a)–(c) above hold with g belonging to $\mathcal{L}^1(R)$ if and only if $H \in B(w,0)$ for some $w \in \mathcal{L}^1(R) \cap \mathcal{L}^\infty(R)$.

We note the following

Corollary 3 Under the conditions of Theorem 4, a linear map $H : U \to M$ is time-invariant.

Theorem 5 provides necessary and sufficient conditions under which the function g defined in (a) of Theorem 4 is continuous.

Theorem 5 Let $H : U \to M$ be linear and time-invariant. Then the following three statements are equivalent:

(i) there is a bounded continuous function $k \in M$ such that

$$(Hu)(t) = \lim_{n\to\infty}\int_{-n}^{n} k(t-\tau)u(\tau)d\tau, \qquad t \in R, \quad u \in U,$$

(ii) $H \in B(w,t)$ for some $w \in \mathcal{L}^\infty(R)$ for each $t \in R$,

$$g(t) \triangleq \lim_{\sigma\to 0}(Hw_\sigma)(t) \tag{10}$$

exists for each $t \in R$, and g is continuous on R, and

(iii) $H \in B(w,t)$ for some $w \in \mathcal{L}^\infty(R)$ for each $t \in R$, and $g(t)$ given by (10) exists for each $t \in R$ with convergence to the limit uniform with respect to t in any bounded interval in R.

References

1. I. W. Sandberg, "Linear Maps and Impulse Responses," IEEE Transactions on Circuits and Systems, CAS-35, No. 2 (February 1988), pp. 201–206.

2. D. Ball and I. W. Sandberg, "g- and h-Representations for Nonlinear Maps," to appear.

3. W. Rudin, Real and Complex Analysis (3rd edition), McGraw-Hill, New York, 1987.

Appendix: Corollary 1-like Results for Maps of Bounded Functions on $[0, \infty)$ or R^m

The following are direct corollaries of Theorems 2 and 3, respectively.

Corollary 4 Let $H : \mathcal{L}^\infty(0,\infty) \to M$ be linear, and let $t \geq 0$. Then the condition that there exists $w \in \mathcal{L}^1(0,\infty) \cap \mathcal{L}^\infty(0,\infty)$ such that $|(Hu)(t)| \leq \|wu\|$ for all $u \in \mathcal{L}^\infty(0,\infty)$ is necessary and sufficient for the following to hold:

(a) $h(t,\tau) \triangleq \lim_{\sigma\to 0} (HT_\tau w_\sigma)(t)$ exists for almost all τ in $[0,\infty)$,

(b) the function $h(t,\cdot)$ belongs to $\mathcal{L}^1(0,\infty) \cap \mathcal{L}^\infty(0,\infty)$, and

(c) we have

$$(Hu)(t) = \int_0^\infty h(t,\tau)u(\tau)d\tau, \quad u \in \mathcal{L}^\infty(0,\infty).$$

Corollary 5 Let $H : \mathcal{L}^\infty(R^m) \to M_m$ be linear, and let $t \in R^m$. Then the condition that there exists $w \in \mathcal{L}^1(R^m) \cap \mathcal{L}^\infty(R^m)$ such that $|(Hu)(t)| \leq \|wu\|$ for all $u \in \mathcal{L}^\infty(R^m)$ is necessary and sufficient for the following to hold:

(a) $h(t,\tau) \triangleq \lim_{\sigma\to 0} (HT_\tau w_\sigma)(t)$ exists for almost all τ in R^m,

(b) the function $h(t,\cdot)$ belongs to $\mathcal{L}^1(R^m) \cap \mathcal{L}^\infty(R^m)$, and

(c) we have

$$(Hu)(t) = \int_{R^m} h(t,\tau)u(\tau)d\tau, \quad u \in \mathcal{L}^\infty(R^m).$$

CLASSICAL AND EXTENDED TRANSMISSION ZEROS: MODEL MATCHING RELATIONSHIPS*

Michael K. Sain
Department of Electrical
and Computer Engineering
University of Notre Dame
Notre Dame, Indiana 46556

Bostwick F. Wyman
Department of Mathematics
The Ohio State University
Columbus, Ohio 43210

Joseph L. Peczkowski
Energy Controls Division
Allied Bendix Aerospace
South Bend, Indiana 46620

ABSTRACT

The module theoretic treatment of the minimal state space, or pole module, of a transfer function for a finite-dimensional linear dynamical system over k-vector spaces leads to finitely generated, torsion k[z]-modules. A corresponding theory was developed by Wyman and Sain for classical transmission zeros in 1981. In the study of solutions P(z) or M(z) to the model matching equation T(z) = P(z)M(z), however, these classical zeros are naturally replaced by certain extended transmission zeros. This paper develops the algebraic relationship between classical and extended transmission zeros, in the context of model matching.

1. INTRODUCTION

Conte, Perdon, and Wyman [1] have recently examined the problem of fixed poles in solutions to the linear transfer function equation

$$T(z) = P(z)M(z), \tag{1-1}$$

where T(z), P(z), and M(z) are linear maps on finite-dimensional vector spaces over the field k(z) of transfer functions having coefficients in a base field k. If R(z), U(z), and Y(z) are the vector spaces, then (1-1) may be seen either in terms of the commutative diagram of Figure 1.1, or that of Figure 1.2, depending upon whether M(z) is given or P(z) is given. For the case in which R(z) = Y(z), and T(z) is the k(z)-linear identity, Wyman and Sain [2,3] obtained earlier results. The problems of Figures 1.1 and 1.2 arise naturally in various applications. In feedback theory,

$$\begin{tikzcd}
& R(z) \arrow[dl, "M(z)"'] \arrow[dr, "T(z)"] & \\
U(z) \arrow[rr, dashed, "P(z)"'] & & Y(z)
\end{tikzcd}$$

Figure 1.1

$$\begin{tikzcd}
& R(z) \arrow[dl, dashed, "M(z)"'] \arrow[dr, "T(z)"] & \\
U(z) \arrow[rr, "P(z)"'] & & Y(z)
\end{tikzcd}$$

Figure 1.2

---

*The work of the first author was supported by The Ohio State University's Distinguished Visiting Professorship, by the University of Notre Dame's Frank M. Freimann Chair in Electrical Engineering, and by the National Science Foundation under Grant ECS-84-05714.

(1-1) has come to be known as the model matching equation. Accordingly, Figures 1.1 and 1.2 represent two versions of the model matching problem.

On the module level, zeros of a composition $P(z)M(z)$ of two given $k[z]$-linear maps have received earlier attention in works of Wyman and Sain [4] and Conte and Perdon [5]. Neither of these, however, addresses the problems where $T(z)$ is given and $M(z)$ or $P(z)$ is the unknown.

In [6], for the case of Figure 1.1, Sain and Wyman have introduced a module of fixed, extended zeros in solutions $P(z)$ to (1-1) and have established a $k[z]$-linear map from this module into the classical transmission zero module of $P(z)$. For this instance, the extended zero differs from the classical zero by failing to be always finitely generated. The authors have continued that study in [7], by characterizing the finitely generated portion of the fixed module of zeros. In a related study [8] last year, Sain and Wyman, for the case of Figure 1.2, developed an analogous module of fixed, extended zeros in solutions $M(z)$ to (1-1). In this case, the extended zero differs from the classical zero by being free instead of torsion. Recently, the authors [9] have developed necessary and sufficient conditions for a solution $P(z)$ or $M(z)$ to (1-1) to have an extended transmission zero module which is isomorphic to the fixed module.

2. POLES AND ZEROS

We set the following conventions. For a field k, let $k[z]$ be the ring of polynomials in z with coefficients in k, and let $k(z)$ be the induced quotient field. Let R, U, and Y be finite-dimensional vector spaces over k; and notice that $k[z]$ and $k(z)$ are also k-vector spaces. Then we employ the k-bilinear tensor product to form $k[z]$-modules

$$R[z] = k[z] \otimes_k R, \qquad U[z] = k[z] \otimes_k U, \qquad Y[z] = k[z] \otimes_k Y, \tag{2-1}$$

and $k(z)$-vector spaces

$$R(z) = k(z) \otimes_k R, \qquad U(z) = k(z) \otimes_k U, \qquad Y(z) = k(z) \otimes_k Y. \tag{2-2}$$

If $M(z) : R(z) \to U(z)$ is a $k(z)$-linear map, we wish to describe its poles and zeros in a precise, technical sense. The pole module $P\bigl(M(z)\bigr)$ associated with a $k(z)$-linear map $M(z)$ is defined to be the $k[z]$-factor module

$$P\bigl(M(z)\bigr) = R[z]/\bigl\{R[z] \cap M^{-1}U[z]\bigr\}, \tag{2-3}$$

where M^{-1} is the inverse image function of $M(z)$, defined on $U[z]$ by

$$M^{-1}U[z] = \bigl\{r(z) \ : \ r(z) \in R(z) \text{ and } M(z)r(z) \in U[z]\bigr\}. \tag{2-4}$$

It is sometimes convenient to make use of the isomorphic form

$$P\bigl(M(z)\bigr) \approx MR[z]/\bigl\{U[z] \cap MR[z]\bigr\}. \tag{2-5}$$

The finitely generated, torsion zero module $Z\bigl(M(z)\bigr)$ associated with a $k(z)$-linear map $M(z)$ was defined by Wyman and Sain [2] in 1981 to be

$$Z\bigl(M(z)\bigr) = \bigl\{M^{-1}U[z] + R[z]\bigr\}/\bigl\{\ker M(z) + R[z]\bigr\}. \tag{2-6}$$

Two isomorphic forms are given by

$$\begin{aligned} Z\bigl(M(z)\bigr) &\approx M^{-1}U[z]/\bigl\{M^{-1}U[z] \cap (\ker M(z) + R[z])\bigr\}, && \text{(2-7)} \\ &\approx \bigl\{U[z] \cap MR(z)\bigr\}/\bigl\{U[z] \cap MR[z]\bigr\}. && \text{(2-8)} \end{aligned}$$

The zeros above were finitely generated and torsion; and, accordingly, they possess a natural interpretation of state-space type. For systems having one input and one output, this concept of zero is entirely adequate. When multiple inputs and multiple outputs are considered, however, zeros appear which are of other types. It is possible, therefore, to extend the idea of zeros.

We begin our discussion of extended zeros with the k[z]-factor module

$$Z_{\Gamma}\big(M(z)\big) = M^{-1}U[z]/\{R[z] \cap M^{-1}U[z]\}. \tag{2-9}$$

This module is torsion. If ker M(z) is not equal to zero, however, (2-9) is not finitely generated. We shall call (2-9) the Γ-zero module of M(z). When ker M(z) is zero, $Z_{\Gamma}\big(M(z)\big)$ is isomorphic, as a k[z]-module, to the zero module $Z\big(M(z)\big)$. When ker M(z) is not zero, the nature of the Γ-zero module can be studied by means of the definition

$$\Gamma\big(M(z)\big) = \ker M(z)/\{R[z] \cap \ker M(z)\} \tag{2-10}$$

and the short exact sequence

$$0 \to \Gamma\big(M(z)\big) \to Z_{\Gamma}\big(M(z)\big) \to Z\big(M(z)\big) \to 0. \tag{2-11}$$

As a divisible module over k[z], $\Gamma\big(M(z)\big)$ is injective; and so its image under injection into $Z_{\Gamma}\big(M(z)\big)$ is a direct summand. It follows that

$$Z_{\Gamma}\big(M(z)\big) \approx \Gamma\big(M(z)\big) \oplus Z\big(M(z)\big). \tag{2-12}$$

We will refer to $\Gamma\big(M(z)\big)$ as the divisible zero module of M(z).

Next we form the k[z]-factor module

$$Z_{\Omega}\big(M(z)\big) = U[z]/\{U[z] \cap MR[z]\}. \tag{2-13}$$

Module (2-13) is finitely generated. Not every equivalence class in (2-13) is torsion, however. $Z_{\Omega}\big(M(z)\big)$ differs from $Z\big(M(z)\big)$ when coker M(z) is nonzero. We call (2-13) the Ω-zero module of M(z).

When coker M(z) is nonzero, form (2-8) is a submodule of (2-13). Identifying (2-8) with $Z\big(M(z)\big)$, we have a natural inclusion of $Z\big(M(z)\big)$ into $Z_{\Omega}\big(M(z)\big)$, with cokernel isomorphic as a k[z]-module to

$$\Omega\big(M(z)\big) = U[z]/\{U[z] \cap MR(z)\}. \tag{2-14}$$

Thus there is a short exact sequence

$$0 \to Z\big(M(z)\big) \to Z_{\Omega}\big(M(z)\big) \to \Omega\big(M(z)\big) \to 0. \tag{2-15}$$

The module (2-14) is torsion-free. It then follows that

$$Z_{\Omega}\big(M(z)\big) \approx \Omega\big(M(z)\big) \oplus Z\big(M(z)\big). \tag{2-16}$$

We will call $\Omega\big(M(z)\big)$ the free zero module of M(z).

3. FIXED EXTENDED ZEROS

Suppose that the dimensions of R, U, and Y are each equal to unity, and consider the scalar equation

$$t(z) = p(z)m(z), \tag{3-1}$$

with all entries in k(z). If we express these transfer functions in relatively prime form

$$t(z) = (n_t(z),d_t(z)),\ p(z) = (n_p(z),d_p(z)),\ m(z) = (n_m(z),d_m(z)), \qquad (3\text{-}2)$$

then we obtain

$$p(z) = (n_t(z)d_m(z),d_t(z)n_m(z)), \qquad (3\text{-}3)$$

which is useful when m(z) is given, and

$$m(z) = (n_t(z)d_p(z),d_t(z)n_p(z)), \qquad (3\text{-}4)$$

which is useful when p(z) is given. From (3-3), we see that the zeros of p(z) consist of those zeros of t(z) which are not zeros of m(z), together with those poles of m(z) which are not poles of t(z). From (3-4), we have that the zeros of m(z) are zeros of t(z) which are not zeros of p(z), together with poles of p(z) which are not poles of t(z).

In order to capture in a more precise way the intuitive ideas of the preceding paragraph, we introduce the notion of a fixed zero module. The fixed Γ-zero module is denoted by Z_Γ and defined as the k[z]-factor module

$$Z_\Gamma = T^{-1}Y[z]/\{T^{-1}Y[z] \cap M^{-1}U[z]\}, \qquad (3\text{-}5)$$

where T(z) : R(z) → Y(z) is a k(z)-linear map. The fixed Ω-zero module is denoted by Z_Ω and defined to be the k[z]-factor module

$$Z_\Omega = \{PU[z] + TR[z]\}/TR[z], \qquad (3\text{-}6)$$

for P(z) : U(z) → Y(z) a k(z)-linear map. The algebraic character of fixed zero modules of Γ-type and of Ω-type is settled by the theorem:

Theorem 1 [7,8]. Let Z_Γ and Z_Ω be the fixed zero modules of Γ-type and Ω-type, respectively, for k(z)-linear maps M(z) : R(z) → U(z), P(z) : U(z) → Y(z), and T(z) : R(z) → Y(z). Define k[z]-factor modules

$$Z'_\Gamma = T^{-1}Y[z]/\{T^{-1}Y[z] \cap (\ker T(z) + M^{-1}U[z])\}; \qquad (3\text{-}7)$$

$$\Gamma = \ker T(z)/\{\ker T(z) \cap M^{-1}U[z]\}; \qquad (3\text{-}8)$$

$$Z'_\Omega = \{PU[z] \cap TR(z)\}/\{PU[z] \cap TR[z]\}; \qquad (3\text{-}9)$$

$$\Omega = PU[z]/\{PU[z] \cap TR(z)\}. \qquad (3\text{-}10)$$

Then

$$Z_\Gamma \approx \Gamma \oplus Z'_\Gamma, \qquad (3\text{-}11)$$

where Γ is torsion divisible; Z'_Γ is finitely generated and torsion; and

$$Z_\Omega \approx \Omega \oplus Z'_\Omega \qquad (3\text{-}12)$$

where Ω is finitely generated and free while Z'_Ω is finitely generated and torsion. Moreover, there are short exact sequences

$$0 \to Z'_\Omega \to Z_\Omega \to \Omega \to 0 \qquad (3\text{-}13)$$

and

$$0 \to \Gamma \to Z_\Gamma \to Z'_\Gamma \to 0. \tag{3-14}$$

So as to justify the use of the term "fixed zeros" in connection with the modules Z_Γ and Z_Ω, we observe that they appear naturally as submodules and factor modules of the appropriate extended zero modules associated with solutions $M(z)$ or $P(z)$ to the model matching equation of Section 1.

Theorem 2 [7,8]. Suppose that Z_Γ and Z_Ω are the fixed zero modules of Γ-type and Ω-type, and let $T(z) : R(z) \to Y(z)$ be a $k(z)$-linear map.

(1) If $P(z) : U(z) \to Y(z)$ is a $k(z)$-linear map whose image contains that of $T(z)$, and if $M(z) : R(z) \to U(z)$ is a $k(z)$-linear map which satisfies the equation $T(z) = P(z)M(z)$, then there exists an epic, $k[z]$-linear map

$$\beta_\Omega(z) : Z_\Omega\big(M(z)\big) \to Z_\Omega, \tag{3-15a}$$

so that Z_Ω is isomorphic to a factor module of the Ω-zero module of $M(z)$.

(2) If $M(z) : R(z) \to U(z)$ is a $k(z)$-linear map whose kernel is contained in that of $T(z)$, and if $P(z) : U(z) \to Y(z)$ is a $k(z)$-linear map which satisfies the equation $T(z) = P(z)M(z)$, then there exists a monic, $k[z]$-linear map

$$\beta_\Gamma(z) : Z_\Gamma \to Z_\Gamma\big(P(z)\big) \tag{3-15b}$$

so that Z_Γ is a submodule of the Γ-zero module of $P(z)$.

Discussion of Theorem: From (3-5), Z_Γ depends upon $T(z)$ and $M(z)$, but not upon $P(z)$; accordingly, the constraint indicated by (3-15b) holds for all solutions $P(z)$ to the model matching equation. Likewise, Z_Ω depends only upon $P(z)$ and $T(z)$; and so (3-15a) is a feature of all solutions $M(z)$.

4. ESSENTIAL SOLUTIONS

When $Z_\Omega\big(M(z)\big) \approx Z_\Omega$, we refer to $M(z)$ as an essential solution to the model matching equation. Likewise, when $Z_\Gamma\big(P(z)\big) \approx Z_\Gamma$, we call $P(z)$ an essential solution. Necessary and sufficient conditions are now given for the existence of essential solutions.

Theorem 3 [9]. Let $T(z) : R(z) \to Y(z)$ be a $k(z)$-linear map. If $P(z) : U(z) \to Y(z)$ is a $k(z)$-linear map whose image contains that of $T(z)$, then there exists a $k(z)$-linear map $M(z) : R(z) \to U(z)$ which satisfies the equation $T(z) = P(z)M(z)$ and which has an Ω-zero module

$$Z_\Omega\big(M(z)\big) \approx Z_\Omega \tag{4-1a}$$

if and only if

$$\operatorname{rank} P(z) - \operatorname{rank} T(z) \geq \dim U - \dim R. \tag{4-1b}$$

If $M(z) : R(z) \to U(z)$ is a $k(z)$-linear map whose kernel is contained in that of $T(z)$, then there exists a $k(z)$-linear map $P(z) : U(z) \to Y(z)$ which satisfies the equation $T(z) = P(z)M(z)$ and which has a Γ-zero module

$$Z_\Gamma\big(P(z)\big) \approx Z_\Gamma \tag{4-2a}$$

if and only if

$$\operatorname{rank} M(z) - \operatorname{rank} T(z) \geq \dim U - \dim Y. \qquad (4\text{-}2b)$$

Discussion of Theorem: The conditions of the theorem, of course, are not always satisfied. This means that essential solutions for model matching problems need not always exist with respect to zeros. Such a situation differs from that in the case of poles. One might ask if, when attention is fixed upon finitely generated, torsion zeros, a result more like that for poles could be obtained. We shall see in the next section that the answer to this question must be given as a negative.

5. CLASSICAL VS. EXTENDED ZEROS

Sections 2-4 have placed a focus upon the extended zero modules of Γ-type and of Ω-type and their role in model matching problems. In this section, our goal is to relate these results to solution zeros of finitely generated, torsion type, and to divisible or free zeros, as explained in Section 3. The main ideas center upon a pair of commutative diagrams.

Theorem 4. Suppose that $T(z) : R(z) \rightarrow Y(z)$ is a $k(z)$-linear map. If $P(z) : U(z) \rightarrow Y(z)$ is a $k(z)$-linear map whose image contains that of $T(z)$, and if $M(z) : R(z) \rightarrow U(z)$ is a $k(z)$-linear map satisfying $T(z) = P(z)M(z)$, then there exist $k[z]$-linear maps $\beta_\Omega(z) : Z(M(z)) \rightarrow Z'_\Omega$ and $\overline{\beta}_\Omega(z) : \Omega(M(z)) \rightarrow \Omega$, the latter an epimorphism, such that the diagram commutes and

$$\begin{tikzcd}
0 \arrow[r] & Z(M(z)) \arrow[r] \arrow[d, "\hat{\beta}_\Omega(z)"'] & Z_\Omega(M(z)) \arrow[r] \arrow[d, "\beta_\Omega(z)"'] & \Omega(M(z)) \arrow[r] \arrow[d, "\overline{\beta}_\Omega(z)"'] & 0 \\
0 \arrow[r] & Z'_\Omega \arrow[r] & Z_\Omega \arrow[r] \arrow[d] & \Omega \arrow[r] \arrow[d] & 0 \\
 & & 0 & 0 &
\end{tikzcd}$$

Figure 5.1

has rows which are short exact sequences. If $M(z) : R(z) \rightarrow U(z)$ is a $k(z)$-linear map whose kernel is included in that of $T(z)$, and if $P(z) : U(z) \rightarrow Y(z)$ is a $k(z)$-linear map satisfying $T(z) = P(z)M(z)$, then there exist $k[z]$-linear maps $\hat{\beta}_\Gamma(z) : \Gamma \rightarrow \Gamma(P(z))$ and $\overline{\beta}_\Gamma(z) : Z'_\Gamma \rightarrow Z(P(z))$, the former a monomorphism, such that the diagram commutes and has rows which are short exact sequences.

$$\begin{tikzcd}
 & 0 \arrow[d] & 0 \arrow[d] & & \\
0 \arrow[r] & \Gamma \arrow[r] \arrow[d, "\hat{\beta}_\Gamma(z)"'] & Z_\Gamma \arrow[r] \arrow[d, "\beta_\Gamma(z)"'] & Z'_\Gamma \arrow[r] \arrow[d, "\overline{\beta}_\Gamma(z)"'] & 0 \\
0 \arrow[r] & \Gamma(P(z)) \arrow[r] & Z_\Gamma(P(z)) \arrow[r] & Z(P(z)) \arrow[r] & 0
\end{tikzcd}$$

Figure 5.2

Proof: Omitted.

Theorem 4 leads to a number of quite detailed consequences. The first of these is a pair of technical corollaries.

Corollary 1. Under the assumptions of Theorem 4, there exist k[z]-linear maps such that the following pair of sequences is exact:

$$0 \to \ker \hat{\beta}_\Omega(z) \to \ker \beta_\Omega(z) \to \ker \bar{\beta}_\Omega(z) \to \operatorname{coker} \hat{\beta}_\Omega(z) \to 0; \tag{5-1}$$

$$0 \to \ker \bar{\beta}_\Gamma(z) \to \operatorname{coker} \hat{\beta}_\Gamma(z) \to \operatorname{coker} \beta_\Gamma(z) \to \operatorname{coker} \bar{\beta}_\Gamma(z) \to 0. \tag{5-2}$$

Proof: Sequences (5-1) and (5-2) are consequences of the Snake Lemma, applied to the pair of diagrams, Figures 5.1 and 5.2.

Corollary 2. Under the assumptions of Theorem 4, the modules in sequences (5-1) and (5-2) are given, up to k[z]-linear isomorphism, by

$$\ker \hat{\beta}_\Omega(z) = \{U[z] \cap (MR[z] + \ker P(z) \cap MR(z))\}/\{U[z] \cap MR[z]\}; \tag{5-3a}$$

$$\ker \beta_\Omega(z) = \{U[z] \cap (MR[z] + \ker P(z))\}/\{U[z] \cap MR[z]\}; \tag{5-3b}$$

$$\ker \bar{\beta}_\Omega(z) = \{U[z] \cap (MR(z) + \ker P(z))\}/\{U[z] \cap MR(z)\}; \tag{5-3c}$$

$$\operatorname{coker} \hat{\beta}_\Omega(z) = \{PU[z] \cap TR(z)\}/\{P[U[z] \cap MR(z)] + PU[z] \cap TR[z]\}; \tag{5-3d}$$

$$\ker \bar{\beta}_\Gamma(z) = M^{-1}\{\ker P(z) + P^{-1}Y[z] \cap U[z]\}/\{\ker T(z) + T^{-1}Y[z] \cap M^{-1}U[z]\}; \tag{5-4a}$$

$$\operatorname{coker} \hat{\beta}_\Gamma(z) = \ker P(z)/\{M(\ker T(z)) + \ker P(z) \cap U[z]\}; \tag{5-4b}$$

$$\operatorname{coker} \beta_\Gamma(z) = P^{-1}Y[z]/\{M(T^{-1}Y[z]) + P^{-1}Y[z] \cap U[z]\}; \tag{5-4c}$$

$$\operatorname{coker} \bar{\beta}_\Gamma(z) = P^{-1}Y[z]/\{M(T^{-1}Y[z]) + \ker P(z) + P^{-1}Y[z] \cap U[z]\}. \tag{5-4d}$$

Proof: The forms follow from standard diagram chases. For brevity, we omit the details.

Theorem 5. Suppose that $T(z) = P(z)M(z)$ is an equation of k(z)-linear maps. Then each of the following three statements implies the others:

$$\Omega(M(z)) \approx \Omega; \tag{5-5a}$$

$$\Gamma(P(z)) \approx \Gamma; \tag{5-5b}$$

$$M(\ker T(z)) = \ker P(z). \tag{5-5c}$$

In case (5-5) holds, then Z'_Ω is a factor module of $Z(M(z))$ and Z'_Γ is a submodule of $Z(P(z))$.

Proof: Condition (5-5a) makes $\ker \bar{\beta}_\Omega(z)$ vanish, which by (5-1) makes coker

$\hat{\beta}_\Omega(z)$vanish, which means that Z'_Ω is a factor module of $Z(M(z))$. A like argument establishes that Z'_Γ is a submodule of $Z(P(z))$. From (5-5c), we find that (5-4b) is zero, which implies (5-5b); again, (5-5c) gives (5-3c) equal to zero, which leads to (5-5a). Now consider (5-5b), which implies that (5-4b) must vanish. If (5-5c) does not hold, then (5-4b) cannot vanish because $\ker P(z) \cap U[z]$ is finitely generated, while $\ker P(z)$ is not. Thus (5-4b) equal to zero must give (5-5c). Finally, we look at (5-5a), for which (5-3c) must vanish. A necessary condition for (5-3c) equal to zero is that

$$U[z] \cap MR(z) \supset U[z] \cap \ker P(z), \qquad (5\text{-}6a)$$

which holds only if

$$U[z] \cap M\big(\ker T(z)\big) \supset U[z] \cap \ker P(z), \qquad (5\text{-}6b)$$

which requires

$$\dim M\big(\ker T(z)\big) \geq \dim \ker P(z); \qquad (5\text{-}6c)$$

and (5-5c) follows.

Example 1 (5-7)

$$T(z) = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}; \; P(z) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & z & 0 \end{bmatrix}; \; M(z) = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

In this example, $M(z)$ is an essential solution; but $P(z)$ is not. On the other hand, conditions (5-5) hold.

Next we turn to a study of the finitely generated, torsion zeros in solutions to the model matching equation.

We begin by directing our attention to the commutative diagram

$$\begin{tikzcd}
U[z] \cap M(\ker T(z) \cap R[z]) \arrow[r] \arrow[d] & U[z] \cap M\big(\ker T(z)\big) \arrow[d] \\
U[z] \cap MR[z] \arrow[r] & U[z] \cap \big(MR[z] + M\big(\ker T(z)\big)\big)
\end{tikzcd}$$

Figure 5.3

of Figure 5.3, wherein the rows and columns are natural inclusions. This diagram induces a $k[z]$-linear map on $Z\big(M(z) \mid \ker T(z)\big)$ into $\ker \hat{\beta}_\Omega(z)$. The kernel of this map is given by

$$\big\{U[z] \cap MR[z] \cap M\big(\ker T(z)\big)\big\} / \big\{U[z] \cap M(\ker T(z) \cap R[z])\big\}. \qquad (5\text{-}8)$$

But this will vanish if

$$M(\ker T(z) \cap R[z]) = M \ker T(z) \cap MR[z], \qquad (5\text{-}9a)$$

which occurs when

$$(\ker T(z) + R[z]) \cap \ker M(z) = \ker T(z) \cap \ker M(z) + R[z] \cap \ker M(z). \qquad (5\text{-}9b)$$

Consider an element in the left member of (5-9b); and represent it by

$$r_t(z) + r(z) = r_m(z). \qquad (5\text{-}10)$$

Applying $T(z)$ to (5-10) shows that $r(z) \in \ker T(z) \cap R[z]$; and so both members of (5-9b) are equal to $\ker M(z)$. The induced map is therefore monic; and $Z\big(M(z) \mid \ker T(z)\big)$ is a submodule of $\ker \hat{\beta}_\Omega(z)$, which is itself a submodule of $Z\big(M(z)\big)$. In this way, we can use $M(z) \mid \ker T(z)$ to design specified zero submodules for solutions $M(z)$. The cokernel of the induced map is

$$\{U[z] \cap (MR[z] + M(\ker T(z)))\}/\{U[z] \cap M(\ker T(z)) + U[z] \cap MR[z]\}. \tag{5-11}$$

Moreover, (5-11) may not disappear; and the solution may display additional zeros. To illustrate, we give an example.

Example 2

$$T(z) = \begin{bmatrix} 1 & 0 \end{bmatrix} \; ; \; P(z) = \begin{bmatrix} 1 & 0 \end{bmatrix} \; ; \; M(z) = \begin{bmatrix} 1 & 0 \\ \frac{1}{z+1} & 1 \end{bmatrix}. \tag{5-12}$$

Here, we have $Z'_\Omega = 0$, $Z(M(z) \mid \ker T(z)) = 0$, and $Z(M(z)) \approx k[z]/(z+1)k[z]$. The solution zero arises entirely from (5-11).

Turning, then, to the possibility of removing some part of Z'_Ω from our solution, we see that decreasing the dimension of $M(\ker T(z))$ is to our advantage. The best we can achieve in this regard is to choose $M(z) \mid \ker T(z)$ equal to zero. In this case, $\ker \hat{\beta}_\Omega(z)$ vanishes, and $Z(M(z))$ is a submodule of Z'_Ω. If the number of non-unit invariant factors of Z'_Ω is no greater than the dimension of $\ker P(z)$, then a solution $M(z)$ with $Z(M(z)) = 0$ is possible. For instance, if the zeros of Z'_Ω are all distinct, then $Z(M(z)) = 0$ is possible whenever $M(z)$ is not unique. Thus there is a tradeoff between inserting into the solution zeros which are distinct from those in Z'_Ω and removing from the solution zeros which are part of Z'_Ω. We summarize in the next theorem.

Theorem 6. Suppose that $T(z) : R(z) \to Y(z)$ is a $k(z)$-linear map. If $P(z) : U(z) \to Y(z)$ is a $k(z)$-linear map whose image contains that of $T(z)$, then $Z(M(z) \mid \ker T(z))$ is a submodule of $Z(M(z))$ whenever a $k(z)$-linear map $M(z) : R(z) \to U(z)$ satisfies the equation $T(z) = P(z)M(z)$. A maximum of

$$\min (\dim \ker P(z), \dim \ker T(z)) \tag{5-13}$$

invariant factors of $Z(M(z))$ can be specified in this way. Also, up to

$$\dim \ker P(z) - \operatorname{rank} M(z) \mid \ker T(z) \tag{5-14}$$

non-unit invariant factors of Z'_Ω can be made into units in $Z(P(z))$ by design of $M(z)$ off the kernel of $T(z)$. If $M(z) : R(z) \to U(z)$ is a $k(z)$-linear map whose kernel is included in that of $T(z)$ and if $P(z)$ is a $k(z)$-linear solution to the equation $T(z) = P(z)M(z)$, then $Z(\overline{P}(z))$ is a factor module of $Z(P(z))$, where $\overline{P}(z)$ is induced on $\operatorname{coker} M(z)$. A maximum of

$$\min(\dim \operatorname{coker} T(z), \dim \operatorname{coker} M(z)) \tag{5-15}$$

invariant factors of $Z(P(z))$ can be specified in this way. In addition, up to

$$\dim \operatorname{coker} M(z) - \operatorname{rank} \overline{P}(z) \tag{5-16}$$

non-unit invariant factors of Z'_Γ can be made into units in $Z(P(z))$ by design of P(z) off the cokernel of M(z).

Proof: The argument for the first part of the theorem has been presented in the prologue. In the interest of space, we omit the second half.

6. CONCLUSIONS

We have studied the constraint relationships imposed upon the classical and extended zeros of k(z)-linear maps $P(z) : U(z) \rightarrow Y(z)$ and $M(z) : R(z) \rightarrow U(z)$ by reason of the fact that they satisfy the model matching equation $T(z) = P(z)M(z)$, in which $T(z) : R(z) \rightarrow Y(z)$ is k(z)-linear as well. This study was inspired by the work of Conte, Perdon, and Wyman [1] on fixed poles in the solution to this same equation. Yet it turns out that the situation for zeros differs markedly from that for poles. Indeed, Theorem 4 indicates that the usual type of zero--which we may think of intuitively as a state-space zero--does not have an intrinsic constraint imposed upon it from its appearance in a solution P(z) or M(z) to the equation. Rather, it is the extended zero, embodying the classical notion of zero together with the novel constructs of divisible or free zeros, which undergoes such constraint. For extensions of the theory to more general rings, see [7,8,10].

REFERENCES

1. G. Conte, A. M. Perdon, and B. F. Wyman, "Fixed Poles in Transfer Function Equations," SIAM Journal on Control and Optimization, Vol. 26, No. 2, pp. 356-368, March 1988.
2. B. F. Wyman and M. K. Sain, "The Zero Module and Essential Inverse Systems," IEEE Transactions on Circuits and Systems, Vol. CAS-27, No. 2, pp. 112-126, February 1981.
3. B. F. Wyman and M. K. Sain, "On the Design of Pole Modules for Inverse Systems," IEEE Transactions on Circuits and Systems, Vol. CAS-31, No. 10, pp. 977-988, October 1985.
4. B. F. Wyman and M. K. Sain, "Exact Sequences for Pole-Zero Cancellation," Proceedings International Symposium on the Mathematical Theory of Networks and Systems, pp. 278-280, August 1981.
5. G. Conte and A. M. Perdon, "Zeros of Cascade Compositions," in Frequency Domain and State Space Methods for Linear Systems, C. I. Byrnes and A. Lindquist, eds. North-Holland: Elsevier, 1986, pp. 23-34.
6. M. K. Sain and B. F. Wyman, "The Fixed Zero Constraint in Dynamical System Performance," Proceedings Eighth International Symposium on the Mathematical Theory of Networks and Systems, C. I. Byrnes, C. F. Martin, and R. Saeks, Editors. North Holland: Elsevier, in press.
7. M. K. Sain, B. F. Wyman, and J. L. Peczkowski, "Zeros in Plant Specification: Constraints and Solutions," Proceedings American Control Conference, June 1988, in press.
8. M. K. Sain and B. F. Wyman, "Extended Zeros and Model Matching," Proceedings Twenty-Fifth Annual Allerton Conference on Communication, Control, and Computing, pp. 524-533, October 1987.
9. M. K. Sain, B. F. Wyman, and J. L. Peczkowski, "Matching Zeros and Multivariable Synthesis," Proceedings IEEE Conference on Decision and Control, December 1988, in press.
10. G. Conte and A. M. Perdon, "An Algebraic Notion of Zeros for Systems over Rings," Proceedings International Symposium on the Mathematical Theory of Networks and Systems, 1983.

An Extension of Kharitonov's Theorem for a Class of Multilinear Coefficient Perturbations

B. Ross Barmish and Zhicheng Shi
Department of Electrical and Computer Engineering
University of Wisconsin–Madison
Madison, Wisconsin 53706

Summary

This short paper gives a necessary and sufficient condition for robust stability of a family of polynomials $\mathcal{P}$ described as follows: Each polynomial $P(s)$ in $\mathcal{P}$ is of the form $P(s) = U(s)V(s) + X(s)Y(s)$ where $U(s), V(s), X(s)$ and $Y(s)$ are interval polynomials. A special case of this result is Kharitonov's Theorem (see [1]) which is obtained when $U(s)$ and $Y(s)$ are taken as unity and $V(s)$ is set to zero.

To motivate the class of multilinearities, consider a rational SISO plant $U(s)/X(s)$ with compensator $V(s)/Y(s)$ where $U(s)$, $V(s)$, $X(s)$ and $Y(s)$ are interval polynomials. Then, analysis of closed loop stability for this system involves studying the zeros of the polynomial

$$P(s) = U(s)V(s) + X(s)Y(s). \tag{1}$$

In view of the motivation above, this paper considers robust stability of polynomials of the form (1) where $U(s)$, $V(s)$, $X(s)$ and $Y(s)$ are interval polynomials.

Definition: $\mathcal{P}$ is said to be *robustly stable* if all polynomials in $\mathcal{P}$ have their zeros in the strict left half plane.

Denote by $\mathcal{U}$ the interval polynomial family associated with $U(s)$ and let $K_i^U(s)$, i=1,2,3,4, be the four *Kharitonov polynomials* for this family. For fixed frequency $w \in \mathbf{R}$, the *cross section* of $\mathcal{U}$ at w is defined by

$$\mathcal{U}_w \doteq \{U(jw) : U(s) \in \mathcal{U}\}.$$

It is well known that this cross section $\mathcal{U}_w$ is a rectangle in complex plane $\mathcal{C}$. Furthermore, by appropriate indexing of the Kharitonov polynomials, this rectangle is the set of complex numbers $U(jw)$ where for each $w \in \mathbf{R}$, the real part $Re\ U(jw)$ satisfies

$$Re\ K_1^U(jw) = Re\ K_4^U(jw) \leq Re\ U(jw) \leq Re\ K_2^U(jw) = Re\ K_3^U(jw). \tag{2}$$

For $w \geq 0$, the imaginary part $Im\ U(jw)$ satisfies

$$Im\ K_1^U(jw) = Im\ K_3^U(jw) \leq Im\ U(w) \leq Im\ K_2^U(jw) = Im\ K_4^U(jw) \tag{3}$$

and for $w < 0$,

$$Im\ K_1^U(jw) = Im\ K_3^U(jw) \geq Im\ U(w) \geq Im\ K_2^U(jw) = Im\ K_4^U(jw). \tag{4}$$

In the same way, for $V(s)$, $X(s)$ and $Y(s)$, we can obtain interval polynomial families $\mathcal{V}$, $\mathcal{X}$ and $\mathcal{Y}$, Kharitonov polynomials $K_i^V(s)$, $K_i^X(s)$ and $K_i^Y(s)$ and cross sections $\mathcal{V}_w$, $\mathcal{X}_w$ and $\mathcal{Y}_w$, respectively.

In order to give a necessary and sufficient condition for the robust stability of $\mathcal{P}$, we proceed as follows: For each fixed angle $\theta \in [0, 2\pi]$ and each fixed frequency $w \in \mathbf{R}$, we define sixteen intervals $I_i(\theta, w)$. Below we describe the construction of $I_1(\theta, w)$. Note that the construction from $I_2(\theta, w)$ through $I_{16}(\theta, w)$ is performed in a similar way; see the full version of this paper [2] for details.

Construction of $I_1(\theta, w)$: For fixed $i \in \{1, 2, 3, 4\}$, we define a lower bound function

$$R_{1i}^-(\theta, w) \doteq \begin{cases} \max\{\frac{Re\ K_1^U(jw)}{\langle K_i^X(jw), e^{-j\theta}\rangle}, 0\} & \text{if } \langle K_i^X(jw), e^{-j\theta}\rangle < 0; \\ 0 & \text{if } \langle K_i^X(jw), e^{-j\theta}\rangle \geq 0 \end{cases}$$

and an upper bound function

$$R_{1i}^+(\theta, w) \doteq \begin{cases} \max\{\frac{Re\ K_1^U(jw)}{\langle K_i^X(jw), e^{-j\theta}\rangle}, 0\} & \text{if } \langle K_i^X(jw), e^{-j\theta})\rangle > 0; \\ 0 & \text{if } \langle K_i^X(jw), e^{-j\theta})\rangle = 0 \text{ and } Re\ K_1^U(jw) \leq 0; \\ +\infty & \text{otherwise.} \end{cases}$$

To complete the construction of $I_1(\theta, w)$, let

$$R_1^-(\theta, w) \doteq \max_{1 \leq i \leq 4} R_{1i}^-(\theta, w)$$

and

$$R_1^+(\theta, w) \doteq \min_{1 \leq i \leq 4} R_{1i}^+(\theta, w).$$

Then define

$$I_1(\theta, w) \doteq (R_1^-(\theta, w), R_1^+(\theta, w)).$$

Now we are in a position to state the main result.

Theorem (see [2] for proof): *$\mathcal{P}$ is robustly stable if and only if the following two conditions are satisfied:*

Condition 1: For each $w \in \mathbf{R}$,

$$0 \notin (\mathcal{U}_w \cup \mathcal{V}_w) \cap (\mathcal{X}_w \cup \mathcal{Y}_w); \tag{5}$$

Condition 2: For each fixed $w \in \mathbf{R}$ and each fixed $\theta \in [0, 2\pi]$, the sixteen intervals $I_1(\theta, w)$, $I_2(\theta, w)$, $\cdots$, $I_{16}(\theta, w)$ cover the set of positive reals; i.e.,

$$\cup_{i=1}^{16} I_i(\theta, w) = (0, +\infty). \tag{6}$$

Note that Condition 1 is easy to check because $\mathcal{U}_w$, $\mathcal{V}_w$, $\mathcal{X}_w$ and $\mathcal{Y}_w$ are rectangles in the two-dimensional complex plane having known vertices involving the real and imaginary parts of the four Kharitonov polynomials. This zero exclusion condition can be checked in many possible ways; even a brute force method will suffice. Covering Condition 2 is also easy to check because we have specific formulae for the endpoints of the intervals $I_i(\theta, w)$.

References

[1] Kharitonov, V. L. (1978). Asymptotic Stability of an Equilibrium Position of a Family of Systems of Linear Differential Equations, *Differential'nye Uraveniya*, Vol. 14, no. 11, pp. 1483 - 1485.

[2] Barmish, B. R., and Z. C. Shi (1988). *Full version of this paper*, Department of Electrical and Computer Engineering Technical Report, No. ECE 88-7, University of Wisconsin-Madison; also submitted for publication.

INVARIANT SUBZEROS ARE NOT INVARIANT*

CHERYL B. SCHRADER and MICHAEL K. SAIN
Department of Electrical and Computer Engineering
University of Notre Dame
Notre Dame, Indiana 46556

ABSTRACT

A linear, time-invariant, finite-dimensional dynamical system may admit subsystems by confining inputs to subspaces or by forming equivalence classes on outputs. Invariant subzeros, which are the invariant zeros of such systems, do not remain invariant under output feedback on the original system.

INVARIANT SUBZEROS

Consider linear time-invariant systems of the form

$$\dot{x}(t) = Ax(t) + Bu(t) \ ; \quad y(t) = Cx(t) + Du(t) \qquad \text{(1a,1b)}$$

where x, u, and y are elements of n, m, and p dimensional vector spaces over R, respectively. The transfer function matrix P(s) corresponding to (1) is a p×m matrix of proper rational transfer functions with real coefficients and having reduced representatives while the Rosenbrock System Matrix $\Sigma(s)$ is a polynomial matrix over R[s] of dimension (n+p)×(n+m). Because it has been previously shown [1] that systems with an unequal number of inputs and outputs rarely have invariant zeros, the case m = p will be considered. The closed-loop response matrix $T(s)=P(s)\left[I+KP(s)\right]^{-1}$ can be expressed in terms of the m×p feedback matrix K over R, assuming the feedback loop is well-posed.

A characterization of invariant zeros of (1), originally defined by H. H. Rosenbrock [2], was given by E. J. Davison and S.-H. Wang as that set of complex numbers $\{s_0\}$ which cause $\Sigma(s_0)$ to drop normal rank [1]. This latter approach can be used to define the invariant zeros of (m-r)×(m-r) subsystems of (1); $1 \leq r \leq m-1$. Example subsystems can be defined by eliminating r columns and r rows from $\Sigma(s)$, producing a Rosenbrock Subsystem Matrix $\Sigma_r(s)$ corresponding to the subsystem

$$\dot{x}(t) = Ax(t) + B_r u_r(t) \ ; \quad y_r(t) = C_r x(t) + D_{rr} u_r(t) \qquad \text{(2a,2b)}$$

where $x(t) \in R^n$, $u_r(t) \in R^{m-r}$, $y_r(t) \in R^{m-r}$. Invariant subzeros of (1), given by the invariant zeros of (2), are designated by that set of complex numbers $\{s_0\}$ which satisfy $\text{rank}\,\Sigma_r(s_0) < n+m-r$. An external description of (2) can be formulated in the customary manner and denoted $P_r(s)$. The transmission zeros of $P_r(s)$ as defined in [3] can be termed transmission subzeros of P(s). Transmission subzeros are a subset of invariant subzeros.

SUBZEROS AND OUTPUT FEEDBACK

The special case when r = 1 can be approached conveniently by using the adjoint as a means of representation. For derivation purposes, the feedback matrix K is assumed to be nonzero only in its first k rows and columns, which are denoted by $\hat{K}$. If K is not in this form, then the inputs and/or outputs can be reordered (i.e., by multiplication with unimodular matrices) to put K into this form. Note that this reordering also affects the original P(s) $\left[T(s)\right]$. Applying the adjoint construction to T(s) yields

$$\mathrm{adj}T(s) = \frac{\left(I+KP(s)\right)}{\det\left(I+KP(s)\right)}\,\mathrm{adj}P(s)\ .$$

Then the dependence of subzeros on output feedback can be summarized by the following theorem.

Theorem: Given the system configuration discussed previously with $\det P(s) \neq 0$, $1 \leq k \leq m$, and a well-posed feedback loop, then

$$\text{(i)}\qquad \left(\mathrm{adj}T(s)\right)_{m-k} = \frac{\left(\mathrm{adj}P(s)\right)_{m-k}}{\det(I_{k,k}+\hat{K}P(s)_{k,k})}\ ,$$

$$\text{(ii)}\qquad \left(\mathrm{adj}T(s)\right)_{k,m-k} = \frac{\left(\mathrm{adj}P(s)\right)_{k,m-k}}{\det(I_{k,k}+\hat{K}P(s)_{k,k})}\ ,$$

$$\text{(iii)}\qquad \left(\mathrm{adj}T(s)\right)_{k,k} = \frac{\left(\mathrm{adj}P(s)\right)_{k,k} + \hat{K}\left(\det P(s)\right)}{\det(I_{k,k}+\hat{K}P(s)_{k,k})}\ .$$

Note that all of the closed-loop minors in (i) - (iii) are related to the corresponding plant minors by $\det\left(I_{k,k}+\hat{K}P(s)_{k,k}\right)$, except those in the upper left k×k block of adjT(s). This is due to the fact that $\hat{K}$ is the upper left k×k block of K. It should be observed, however, that the upper left block of adjT(s) does not translate into the upper left block of T(s).

From (i) it can be seen that the elements of the last m-k rows of adjP(s) have numerators which can possibly be affected by cancellation or addition of factors, but their zeros cannot be shifted as a result of nonzero output feedback. Similarly, (ii) shows that this holds true for the first k rows and the last m-k columns of adjP(s). Next consider (iii), a description of the first k rows and the first k columns of adjT(s). The only element zeros which can possibly be shifted in adjP(s) are those in positions where $\hat{K}$ is nonzero. The number of (m-1)×(m-1) subtransmissions whose zeros may shift with respect to a change in $\hat{K}$ is equal to the number of nonzero entries in $\hat{K}$ (modulo $(\det(I_{k,k}+\hat{K}P(s)_{k,k}))$.

Subzero shifting can be used to precondition a plant to meet design specifications on transient response or reduce control difficulties due to dominant time-frequency frame zeros. For a more detailed discussion and proofs, see [4].

REFERENCES

1. E. J. Davison and S.-H. Wang, "Properties and Calculation of Transmission Zeros of Linear Multivariable Systems," Automatica, Vol. 10, No. 6, pp. 643-658, December 1974.
2. H. H. Rosenbrock, "The Zeros of a System," International Journal of Control, Vol. 18, No. 2, pp. 297-299, August 1973.
3. H. H. Rosenbrock, State-space and Multivariable Theory. New York: John Wiley & Sons, Inc., 1970.
4. Cheryl Bunnett Schrader, "Subzeros of Linear Multivariable Systems," M.S. Thesis, Department of Electrical and Computer Engineering, University of Notre Dame, Notre Dame, Indiana, April 1987.

---

[*] This work was supported in part by the National Science Foundation under Grant ECS-84-05714 and by the University of Notre Dame's Frank M. Freimann Chair in Electrical Engineering. The work of the first author was also supported by the SAE Forgivable Loan Program; that of the second author also received support from The Ohio State University's Distinguished Visiting Professorship.

ON STABLE SOLUTIONS OF THE ONE AND TWO SIDED MODEL MATCHING PROBLEMS

Zhiqiang Gao and Panos J. Antsaklis
Department of Electrical and Computer Engineering
University of Notre Dame
Notre Dame, IN 46556

Abstract

An algorithm is introduced to determine proper and stable solutions to the Model Matching Problem. It utilizes the theory of inverses and state-space algorithms, and it guarantees a proper and stable solution to the problem when one exists. This approach is also used to determine proper and stable solutions of the Two-Sided Matching Problem. Examples are included.

I. Introduction

The Model Matching Problem (MMP) is defined as follows:
Given proper rational matrices P(pxm) and T(pxq), find a proper rational matrix M such that the equation

$$PM = T \tag{1}$$

holds. This problem is also being referred to as the Exact Model Matching Problem.

The Model Matching Matching Problem has drawn the attention of many researchers both because of its importance in control and its attractive mathematical formulation. Note that equation (1) has also been studied in the literature over polynomials, rationals, and rings using different mathematical tools and methods.

The MMP was formulated and proposed by Wolovich [1] in the early 70's. Its significance in control problems is well known. Here the main interest is in the case where P is proper and T is proper and stable, and a proper and stable M is to be determined. We call this problem Stable MMP (SMMP). Although the conditions for existence of proper and stable solutions are known[10,15,16], it appears that there is no satisfactory approach to determine the solution when one exists. It should be mentioned that the algorithms which have been reported in the literature[6,7] exhibit unsatisfactory performance. Early attempts used linear state feedback realizations of M to solve a restricted version of the problem. In particular, it was early recognized that a state feedback controller can be presented as a mathematically equivalent open loop control law M of the form PM = T. This led to attempts to solve a restricted form of the SMMP (T constrained) solvable via some state feedback "equivalent" M [1,2]; M in this case was proper and stable and it corresponded to a stabilizing stable feedback control law. The MMP was later formulated as a kernel problem[3,4,12] in which the stability of solutions was ignored; in [5] the properness of solutions was not considered. In this paper an algorithm is presented to determine proper and stable solutions of the SMMP.

Assume that a proper and stable solution M to (1) does exist. The interest here is in the common in control cases where rank[P] = p ≤ m. In this case there exists a right inverse P_{ri} of P, so that $PP_{ri} = I_{pxp}$. Clearly

$$M = P_{ri}T \tag{2}$$

satisfies (1), and it is shown below that M can always be chosen to be proper and stable. The advantage of this approach is that the extensive results on inverses, which include algorithms in state space, can be used here with few modifications; furthermore after P_{ri} has been found, solutions M for different T's can be easily calculated. Note that in many problems T is not really fixed but it can vary within certain limits [10]. In many cases the specifications of the problem allow us to choose T so that a proper and stable solution M to (1) does exist. A method for choosing such T is given in [10]; see also C in Appendix. Here it is assumed that P and T are such that a proper and stable solution M to SMMP does exist, and this is the starting point in our approach.

The approach to solve SMMP is subsequently used to determine proper and stable solutions to the Two-Sided Matching Problem (TSMP). The Two-Sided Matching Problem (TSMP)[8,9] arises in the multivariable control synthesis problem of designing a controller which makes the outputs of a physical system respond in a desirable manner to reference inputs and disturbances. In particular, consider the closed-loop system where S is the system to be controlled:

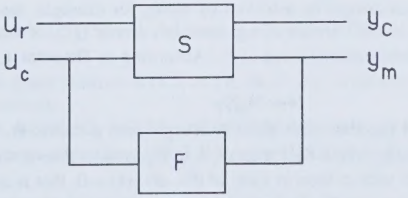


Figure 1

Let

$$\begin{bmatrix} y_c(s) \\ y_m(s) \end{bmatrix} = \begin{bmatrix} G(s) & H(s) \\ M(s) & N(s) \end{bmatrix} \begin{bmatrix} u_c(s) \\ u_r(s) \end{bmatrix} \tag{3}$$

where u_r is the vector of reference and disturbance inputs, u_c is the vector of control inputs, y_m is the available measurement output vector, y_c is the vector of outputs to be controlled, and G, H, M and N are rational matrices. Let the controller be

$$u_c = Fy_m$$

so that the closed-loop system of Figure 1 is stable and a desired transfer function H_d is obtained between u_r and y_c. Through a simple calculation,

$$y_c = \{GF[I - MF]^{-1}N + H\}u_r = H_du_r$$

Now the problem can be formulated as follows:
Given G, H, M, N and H_d, find a proper controller F so that

i) The closed loop system is stable, and

ii) $GF[I\text{-}MF]^{-1}N+H = H_d$ (4)

This is the so called General Servomechanism Problem (GSP) [8]. Define

$$X := F[I\text{-}MF]^{-1}$$

so that

$$F = [I+XM]^{-1}X$$

X must be stable for internal stability. Substitute F in (4) to obtain

$$GXN + H = H_d$$

or

$$GXN = H_d - H \tag{5}$$

We call this type of problem the Two-Sided Matching Problem (TSMP). In general, the TSMP is defined as:
Given the rational matrices A, B, and C, find X such that:

$$AXB = C \tag{6}$$

We are interested in proper and stable solutions X where A, B are proper and C is proper and stable. The approach used for the SMMP can be directly applied to determine proper and stable solutions of the TSMP. Based on the method presented below, a solution X is found of the form

$$X = A_{ri}CB_{li} \tag{7}$$

where A_{ri} and B_{li} are the corresponding right and left inverses of A and B.

In Section II, the existence of a proper and stable solution of the SMMP is discussed. In Section III, an algorithm to determine such solutions and an example are given. In Section IV, it is explained how to apply the algorithm to solve the TSMP. An example in [8] is used to compare our results with the previous ones. Finally in Section V, some concluding remarks are included.

II. Preliminaries

Consider (1) and the SMMP problem. For a proper and stable solution M to exist, P and T have to satisfy certain conditions. We give the following existence theorem without proof:

Theorem 1[10]: Given P proper, T proper and stable with rank[P] = rank[T] = p, there exist proper and stable solutions M if and only if T has as its zeros all the RHP finite zeros and all the zeros at infinity of P together with their associated structure (described below).

Note that similar results have appeared elsewhere in the literature, e.g. [15,16]. The theorem and its significance is discussed below:
Let $P = ND^{-1}$ and $T = N_TD_T^{-1}$ be right coprime polynomial fraction descriptions of P and T, and write

$$N = N_b\underline{N} \tag{8}$$

where the pxp matrix N_b contains all the RHP zeros of P in the sense that the roots of $|N_b|$ ($\neq 0$) are exactly the RHP zeros of P. This can always be achieved by using, for example, the Smith form of N. Note that since P has full row rank, N_b is a left divisor of a greatest left divisor (gld) of the rows of N, the determinant of which has roots the finite (transmission) zeros of P. According to Theorem 1, for a stable solution to exist, N_T has to have the form

$$N_T = N_b\underline{N}_T \tag{9}$$

That is the RHP zeros of P together with their structure (zero directions), which appear in N_b, appear unchanged in T. To illustrate, let z be a RHP zero of P, in N_b, and let the vector a satisfy $aN_b(z) = 0$, that is a gives the direction associated with z; then in view of (9), $aN_T(z) = 0$, that is z together with its direction a ((z,a) imposes certain restrictions on the structure of N_T, via $aN_T(z) = 0$) will appear in T. For the zeros at infinity, the same argument is true. This can be proved in the same way by substituting $s = 1/w$ for s and studying the zeros at $w = 0$.

Note that Theorem 1 is also valid when $\mathrm{rank}[T] < p$ with slight modifications. In this case the solvability condition is also relation (9), which however does not necessarily imply that the RHP zeros of P in N_b are also zeros of T.

III. Main Results

The solutions derived here are of the form $M = P_{ri}T$ (2). It is first shown that solutions M can always be chosen to be proper and stable when such solutions exist. Then an algorithm is given to derive such M.

Lemma 1: Let $PP_{ri} = I_{pxp}$, then P_{ri} can always be written as:

$$P_{ri} = N_i(N_b\underline{D}_i)^{-1} \tag{10}$$

where N_i and $(N_b\underline{D}_i)$ are right coprime polynomial matrices with N_b defined in (8) and $\underline{D}_i$ a polynomial matrix.

Proof: Let $P = ND^{-1}$ and $P_{ri}= N_iD_i^{-1}$ be right coprime polynomial matrix fraction descriptions of P and P_{ri}, then

$$ND^{-1}N_iD_i^{-1}= I$$

which implies that

$$ND^{-1}N_i = D_i \tag{11}$$

Clearly the left hand side is a polynomial matrix and all the poles of P in D cancel with N_i since N and D are right coprime. Substitute (8) to (11)

$$N_b(\underline{N}D^{-1}N_i) = D_i$$

Therefore D_i can be written as

$$D_i = N_b\underline{D}_i \tag{12}$$

where $\underline{D}_i = \underline{N}D^{-1}N_i$ is a polynomial matrix. Q.E.D

Remark: Lemma 1 shows that all the RHP zeros of P, in N_b, will appear as poles of any right inverse of P. This is of course a well known result. The lemma however shows more than this. In particular, the contribution here is that the lemma shows that not only the RHP zero locations of P will be poles of P_{ri}, but also the associated RHP zero structure (zero directions) of P will appear in the pole structure of P_{ri}.

Theorem 2: If Theorem 1 is satisfied and P_{ri} does not introduce any new RHP poles other than RHP zeros of P, then M in (2) is proper and stable.

Proof: In view of Theorem 1 and (9), $P = N_b\underline{P}$ and $T = N_b\underline{T}$. Since N_b is nonsingular, $\underline{P}$ still have full row rank. From (2) and Lemma 1,

$$M = P_{ri}T = (\underline{P}_{ri}N_b^{-1})(N_b\underline{T}) = \underline{P}_{ri}\underline{T}$$

Since $\underline{T}$ is stable and $\underline{P}_{ri}$ contains only the stable poles of P_{ri}, M is stable. The properness of M can be proven by substituting $s = 1/w$. The zeros of P(w) and T(w) at $w = 0$ are the zeros of infinity of P(s) and T(s) and by using similar approach, it is shown that M has no poles at $w = 0$. Q.E.D

The algorithm introduced below to derive such proper and stable M utilizes results from the literature on inverses. See [11] for a survey. Here we are only interested in those algorithms which can arbitrarily assign the nonfixed poles of the inverse system. Antsaklis[13] gives a simple stable inverse algorithm but it works only for plants P where $\lim_{s\to\infty} P$ is of full rank. Patel[14] extended the results to also apply to strictly proper plants. In the algorithm introduced below, we combine the merits of both inverse algorithms to solve SMMP(see A, B in Appendix).

Given P with $\mathrm{rank}[P] = p$, a right inverse P_{ri} is first determined. The only RHP poles of P_{ri} are the RHP zeros of P. In Step 5 a solution M of SMMP is determined as $M = P_{ri}T$. It is proper and stable, in view of above result.

The SMMP Algorithm:

Step1 Find an irreducible state space realization of P as {A, B, C, E}, where A, B, C, and E are nxn, nxm, pxn and pxm real matrices respectively.

Step2 If $\mathrm{rank}[E] = p$, set $A' = A$, $B' = B$, $C'_q = C$, $E'_q = E$ and $\Phi(s) = I$, and go to step4.

If $\mathrm{rank}[E] < p$, find C_q, E_q and $\Phi(s)$ such that $\mathrm{rank}[E_q] = p$ and $P = \Phi^{-1}(s)\,[C_q(sI - A)^{-1}B + E_q]$ where $\Phi(s)$ is pxp nonsingular polynomial matrix.

Step3 Find $\{A', B', C'_q, E'_q\}$, an irreducible representation of $\{A, B, C_q, E_q\}$.

Step4 Find a proper right inverse of the system $\{A', B', C'_q, E'_q\}$ as $\{A_I, B_I, C_I, E_I\}$. Calculate the transfer matrix $P_I = C_I(sI-A_I)^{-1}B_I+E_I$.
Step5 Calculate $M = P_{ri}T = P_I\,\Phi(s)T$ which is a solution of the SMMP.

Remark1. For Step1, many algorithms are available, e.g. the Hankel method[12], or the Coprime Fraction method[3,12].
Remark2. In Step 2, a method in [14] can be used to find C_q, E_q and $\Phi(s)$. It is also outlined in Appendix.
Remark3. In Step 4, we can use the algorithm in [13](see Appendix). Assume P is of order n and it has k zeros, then a proper nth-order right inverse can always be found with k zeros of P as its poles and remaining (n-k) poles be arbitrarily assignable.
Remark4. Note that this algorithm can be directly used in the more general case of Ω - stabilization, considered in [15], although here we concentrate only in the standard stabilization problems (no poles in the RHP). To do so the location of the zeros included in N_b must be changed from the RHP to Ω^c, the complement of Ω, which is a (good) region of the complex plane symmetric with respect to the real axis.

Example 1:
Let

$$P = \begin{bmatrix} \frac{s-1}{s(s+1)} & \frac{s-1}{s(s+2)} \end{bmatrix} = [0\ ,\ s-1]\begin{bmatrix} -(s+1) & s(s+1) \\ s+2 & 0 \end{bmatrix}^{-1}$$

a right coprime fraction.
Step1: An irreducible realization of P is

$$A = \begin{bmatrix} -2 & 0 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & -1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}, \quad C = [0\ \ -1\ \ 1],\ \ E = [0\ \ 0]$$

Step2: Since rank[E] = 0 < p, we apply the method in [14] and find

$$C_q = [-1\ \ 0\ -2] \qquad E_q = [1\ \ 1] \quad \text{and} \quad \Phi(s) = s$$

Step3: An irreducible representation of the system $\{A, B, C_q, E_q\}$ is

$$A' = \begin{bmatrix} -1 & 0 \\ 0 & -2 \end{bmatrix} \quad B' = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad C'_q = [-2\ \ -3] \quad E'_q = [1\ \ 1]$$

Step4: Using the algorithm in [13], we find

$$A_I = \begin{bmatrix} -2 & 0 \\ 3 & 1 \end{bmatrix} \quad B_I = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad C_I = \begin{bmatrix} -2 & 0 \\ 3 & 3 \end{bmatrix} \quad E_I = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{and}$$

$$P_I = C_I(sI-A_I)^{-1}B_I+E_I = \begin{bmatrix} \frac{(s+1)}{(s+2)} \\ \frac{3}{(s-1)} \end{bmatrix} \qquad P_{ri} = P_I\Phi(s) = \begin{bmatrix} \frac{s(s+1)}{(s+2)} \\ \frac{3s}{(s-1)} \end{bmatrix}$$

Step5: If T is chosen as:

$$T = \begin{bmatrix} \frac{(s-1)}{(s+1)(s+3)} & \frac{(s-1)}{(s+1)(s+4)} \end{bmatrix}$$

then a proper and stable M to PM = T is

$$M = P_{ri}T = \begin{bmatrix} \frac{s(s-1)}{(s+2)(s+3)} & \frac{s(s-1)}{(s+2)(s+4)} \\ \frac{3s}{(s+1)(s+3)} & \frac{3s}{(s+1)(s+4)} \end{bmatrix}$$

Notice that P_{ri} is not proper.

IV. Application to the Two-Sided Matching Problem

The basic idea is to use the stable inverse algorithm, that was used for SMMP, to solve the Two-Sided Matching Problem(TSMP). We believe that this approach has certain advantages over the earlier results by Ohm etc.[8] and by Ozguler etc.[9]. First, it not only gives the conditions on the existence of solution but also gives a computational method to find the solution. Second, it provides some insight into the problem in that the existence of proper and stable solution X depends on the locations and directions of the RHP and infinite zeros of the transfer matrices A, B and C. This leads to a practical approach to pick C which is related to the closed-loop transfer matrix in view of (5). To illustrate the approach consider (5), and suppose that z is a zero of G (or N), and the row vector a is determined from aG(z) = 0 (or aN(z) = 0), then according to the theory in [10](see C in Appendix), the zero z together with its structure will appear in (H_d-H) if and only if

$$a(H_d(z) - H(z)) = 0$$

or

$$aH_d(z) = aH(z) \qquad (13)$$

It appears this method gives more clear constraints on H_d and is more computationally efficient than the results in [8] and [9].

Consider

$$AXB = C \qquad (6)$$

Theorem 3: Suppose C is proper and stable, A and B are proper and have full row rank and full column rank respectively. Then the TSMP in (6) has a proper and stable solution if C has all the RHP zeros and infinite zeros of A and B together with their associated structures (in the sense of Theorem 1).

Proof: The stability of solution X is proved below; the properness can be proved by using the same variable substitution, s=1/w, as in SMMP.

Consider solutions of the form:

$$X = A_{ri}CB_{li} \qquad (14)$$

Let $A = N_{b1}\underline{N}_A D_A^{-1} = N_{b1}\underline{A}$ and $B = D_B^{-1}\underline{N}_B N_{b2} = \underline{B}N_{b2}$, where N_{b1} and N_{b2} are similarly defined as N_b in (8), then if C satisfies the condition of the theorem it can be factorized as

$$C = N_{b1}\underline{C}N_{b2} \qquad (15)$$

with $\underline{C}$ an appropriate stable transfer matrix. By Lemma 1, a solution will be

$$X = (\underline{A}_{ri}N_{b1}^{-1})(N_{b1}\underline{C}N_{b2})(N_{b2}^{-1}\underline{B}_{li}) = \underline{A}_{ri}\underline{C}\underline{B}_{li} \qquad (16)$$

Since the inverse algorithm does not introduce any RHP zeros, $\underline{A}_{ri}$ and $\underline{B}_{li}$ will be stable. Therefore X will also be stable. Q.E.D

Note that Theorem 3 is valid regardless of the row or column rank of C with slight modifications. When C does not have full row or column rank, the solvability condition is also (15), where however the zeros contained in N_{b1} and N_{b2} are not necessarily the zeros of C.

Example 2:

A simple single-input single-output example from [8] is used to compare our results to the known results in literature, although the method suggested above shows better performance on multi-input multi-output system. Given the system in Figure 2, characterize all admissible H_d.

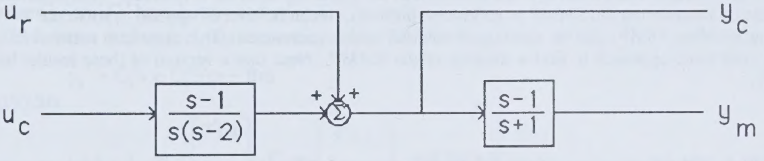


Figure 2

The system matrix is

$$\begin{bmatrix} G(s) & H(s) \\ M(s) & N(s) \end{bmatrix} = \begin{bmatrix} \dfrac{-(s-1)}{s(s-2)} & 1 \\ \dfrac{-(s-1)^2}{s(s+1)(s-2)} & \dfrac{(s-1)}{(s+1)} \end{bmatrix}$$

Let H_d be the desired transfer function from u_r to y_c. We want to characterize all the H_d such that there exist proper and stable solutions X of the equation

$$GXN = H_d - H$$

Here A = G, B = N, $C = H_d - H$, and $N_{b1} = N_{b2} = (s-1)$. According to Theorem 3, for a proper and stable solution X to exist, C must have the form

$$C = (s-1)^2\underline{C}$$

where $\underline{C}$ must be proper and stable with at least one infinite zero. Considering in this example H = 1 and H_d is a scalar transfer function, C = (n(s) - s(s))/d(s), where H_d = n(s)/d(s), and it is straight forward to find the following constraints on $H_d(s)$:

1) d(s) must be a Hurwitz polynomial,
2) n(s) and d(s) must satisfy

$$n(s) - d(s) = (s-1)^2 k(s)$$

for some polynomial k(s), and
3) The relative degree of (n(s) - d(s)) and d(s) must be greater than zero.

From these constraints we can see that d(s) is almost an arbitrary Hurwitz polynomial of degree higher than two, while n(s) can be obtained from the equation

$$n(s) = (s-1)^2 k(s) + d(s)$$

where k(s) is chosen to make n(s) close to the desired zero polynomial and relative degree of (n(s)-d(s))/d(s) greater than zero. Suppose d(s) is chosen to be d(s) = (s+1)(s+2)(s+3), and n(s) is only required to have its zeros in left half plane, then k(s) can be chosen as k(s) = 1, and therefore n(s) = $s^3 + 7s^2 + 9s + 7$, which has all its roots in LHP. The solution is

$$X = G^{-1}(H_d - H)N^{-1} = \frac{-s(s-2)}{(s+2)(s+3)}$$

and the compensator F is

$$F = [I+XM]^{-1}X = \frac{-s(s+1)(s-2)}{n(s)}$$

which is proper and stable.

Comparing this example with the one in [8], we can see the method proposed here has several advantages. First it is more intuitive since it gives a direct relation between poles and zeros of G, N, H_d and H and the existence of proper and stable solution X. Second, the constraints on H_d are fairly easy to apply when we try to pick an H_d that not only guarantees the existence of a proper and stable compensator F, but also meets the closed-loop performance specifications.

IV. Conclusions

In this note, we presented a new algorithm to solve the Stable Model Matching Problem(SMMP). It is guaranteed that the solution will be stable if the condition in the existence theorem is fulfilled. This algorithm is made possible by formulating the SMMP as an inverse problem. It can be directly applied to solve the Two-Sided Matching Problem(TSMP) and the stability of solution is also guaranteed. This algorithm is believed to be a practical systematic approach to find a solution of the SMMP. Note that a version of these results has appeared in [17].

References

[1] Wolovich, W.A., "The Application of State Feedback Invariants to Exact Model Matching," presented at the 5th Annual Princeton Conf. on Information Sciences and Systems, Princeton, N.J., March 1971.

[2] Moore, B.C. and Silverman, L.M., "Model Matching by State Feedback and Dynamic Compensation," IEEE Trans. Automatic. Control., Vol. AC-17, No.4, pp. 491-497, August 1972.

[3] Wolovich, W.A., Linear Multivariable Systems, Springer-Verlag, New York, 1974.

[4] Forney, G.D. Jr., "Minimal Bases of Rational Vector Space with Applications to Multivariable Linear Systems," SIAM J. Contr., Vol. 13, pp. 493-520, May 1975.

[5] Wolovich, W.A., Antsaklis, P.J., and Elliott, H., " On the Stability of Solutions to Minimal and Non-Minimal Design Problems," IEEE Trans. Automatic Control, Vol. 22, pp. 88-94, February 1977.

[6] Scott, R.W. and Anderson, B.D.O., "Least Order Stable Solution of the Exact Model Matching Problem." Automatica, Vol.14, pp. 481-492, September 1978.

[7] Scott, R.W. and Anderson, B.D.O., "Parameter Solution of the Stable Exact Model Matching Problem," IEEE Trans. Automatic Control, AC-22, pp. 137-138, June 1977.

[8] Ohm, D.Y., Howze, J.W. and Bhattacharyya, S.P., "Structural Synthesis of Multivariable Controllers," Automatica, Vol.21, No.1, pp. 35-55, January 1985.

[9] Ozguler, A.B. and Eldem, V., " Disturbance Decoupling Problems Via Dynamic Output Feedback," IEEE Trans. Automatic Control, Vol. AC-30, No. 8, pp. 756-764, August 1985.

[10] Antsaklis, P.J., "On Finite and Infinite Zeros in the Model Matching Problem," Proc. of the 25th IEEE Conf. on Decision and Control, Athens Greece, December 10-12, 1986.

[11] Wyman, B.F. and Sain, M.K., "On the Design of Pole Modules for Inverse Systems," IEEE Trans. Circuits Syst., Vol. CAS-32, No.10, pp. 977-988, October 1985.

[12] Chen, C.T., Linear System Theory and Design, Holt, Rinehart and Winston, New York, 1984.

[13] Antsaklis, P.J., " Stable Proper nth-order Inverses," IEEE Trans. Automat. Contr., Vol. AC-23, pp. 1104-1106, December 1978.

[14] Patel, R.V., "Construction of Stable Inverses for Linear Systems", Int. J. Systems Sci., Vol.13, No.5, pp. 499-515, 1982.

[15] Vardulakis, A.I.G. and Karcanias, N., "On the Stable Exact Model Matching Problem," System & Control Letters, pp. 273-242, February 1985.

[16] Pernebo, L., "An Algebraic Theory for the Design of Controllers for Linear Multivariable System, Part I, II," IEEE Trans. Automatic Control, AC-26, pp. 171-194, February 1981.

[17] Gao, Z., "The Model Matching and Related Control Problems:Conditions and Algorithms," M.S. Thesis, Dept. of Electrical and Computer Engineering, Univ. of Notre Dame, August 1987.

Appendix

A. Solving Inverse Problem of Strictly Proper Plant in State Space[14]

Assume the plant P we are working on has the state space form {A, B, C, E}. The algorithm in [13] only works on those with E has full row rank. In order to use this algorithm for any plant {A, B, C, E}, we use the following algorithm[14] to find P in the form

$$P = \Phi^{-1}(s)[C_q(sI - A)^{-1}B + E_q] \tag{A-1}$$

where $\Phi(s)$ is pxp nonsingular polynomial matrix, E_q has full row rank.

Algorithm:

Initialization: set $i = 0$, $y_i = y$, $C_i = C$ and $E_i = E$.

Step i

(I) Construct an pxp matrix U_i to compress the rows of E_i

$$U_iE_i = \begin{bmatrix} E_i' \\ 0 \end{bmatrix} \tag{A-2}$$

where E_i' has full row rank and $l_i = \text{rank}\,[E_i']$ is defined by the compression.

(II) If $l_i = p$ then go to (V).

If $l_i < p$, partition U_iy_i and U_iC_i conformably with the compression of E_i

$$U_iy_i = \begin{bmatrix} y_i' \\ y_i'' \end{bmatrix} \quad \text{and} \quad U_iC_i = \begin{bmatrix} C_i' \\ C_i'' \end{bmatrix} \tag{A-3}$$

where y_i' and C_i' have l_i rows and $y_i''$and C_i'' have $(p - l_i)$ rows.

(III) Replace the last $p - l_i$ equations of (A-3) by the following

$$\dot{y}_i'' = C_i'\dot{x} = C_i'(Ax + Bu) \tag{A-4}$$

(IV) Set

$$i = i + 1,\; y_i = \begin{bmatrix} y'_{i-1} \\ \dot{y}''_{i-1} \end{bmatrix},\; C_i = \begin{bmatrix} C'_{i-1} \\ C''_{i-1}A \end{bmatrix},\; \text{and } E_i = \begin{bmatrix} E'_{i-1} \\ C''_{i-1}A \end{bmatrix}, \tag{A-5}$$

Note the new output equation is

$$y_i = C_ix + E_iu \tag{A-6}$$

Go to step i.

(V) Set $q = i$. C_q and E_q are obtained with $\text{rank}[E_q] = p$, and the relation between y_q and $y_0 (= y)$ can be expressed as

$$y_q = S_{q-1}(p)U_{q-1}S_{q-2}(p)U_{q-2}\ldots S_0(p)U_0y = \Phi(p)y \tag{A-7}$$

where

$$S_i(p) = \begin{bmatrix} I_{l_i} & 0 \\ 0 & I_{(p-l_i)}p \end{bmatrix},\quad p = \frac{d}{dt}, \text{ and}$$

$$\Phi(p) = S_{q-1}(p)U_{q-1}S_{q-2}(p)U_{q-2}\ldots S_0(p)U_0$$

The transfer function matrix of (A,B,C_q,E_q) is given by

$$P_q = C_q(sI_n - A)^{-1} + E_q = \Phi(s)[C(sI_n - A)^{-1} + E] = \Phi(s)P \tag{A-8}$$

The inverse of P_q can be derived using the algorithm below.

B. Stable Inverse Algorithm[13]

Let $E = \lim_{s \to \infty} P$ satisfying rank[E] = p. Assume P is of order n and it has k zeros, then a proper nth-order right inverse can always be found with k zeros of P as its poles and remaining (n-k) poles be arbitrarily assignable. Suppose that the state space realization of P is { A, B, C, E }, the following procedure is designed to determine the inverse of P with (n-k) non-fixed poles stable.

i). Find a mxm nonsingular matrix M such that $EM = [I_p : 0]$.

ii). Calculate $[\, \underline{B}_1, \underline{B}_2 \,] = BM$ and $A - \underline{B}_1 C$.

iii). Find a lsvf matrix $\underline{F}_2$ which assigns the n-k controllable poles of $\{A-\underline{B}_1 C, \underline{B}_2\}$ in the LHP.

iv). The desired proper right inverse is

$$\left\{ A + BM\begin{bmatrix} -C \\ \underline{F}_2 \end{bmatrix},\ BM\begin{bmatrix} I_p \\ \underline{G}_2 \end{bmatrix},\ M\begin{bmatrix} -C \\ \underline{F}_2 \end{bmatrix},\ M\begin{bmatrix} I_p \\ \underline{G}_2 \end{bmatrix} \right\}$$

where $\underline{F}_2$ was determined above and $\underline{G}_2$ any (m-p)xp real matrix (which can be taken as 0 for convenience).

C. Selecting T in Control Design[10]

In control, T in T = PM is chosen so that the system response y = Tr to test input satisfies the control design specifications. The relation $N_T = N_b \underline{N}_T$ which characterizes the unstable finite zeros and the zeros at infinity T must have for a proper and stable solution M to exist, does not provide a convenient way to choose an appropriate T. Note that the transfer function entries in T are individually chosen to satisfy specifications, and their zeros do not necessarily appear as zeros of T. Therefore there is a need for simple and direct conditions which will help the designer to choose T containing the unavoidable unstable or infinite zeros together with their directions. The following Theorem on unstable zeros is given below without proof; the results may apply to infinite zeros with slight modifications.

Assume that z_i, i = 1,..., l, are distinct or if z_j is a multiple zero the rank reduction in $N(z_j)$ equals the multiplicity of z_j, where $P = ND^{-1}$ is in a coprime polynomial fraction form.

Theorem A: The unstable zeros of P together with their structure will appear in T if and only if

$$a_i T(z_i) = 0 \qquad \text{(A-9)}$$

where a_i is determined from

$$a_i P(z_i) = 0 \qquad \text{(A-10)}$$

For a simple example see Example 1, where P has only one zero at $z_1 = 1$. Here a_1 could be an arbitrary constant since $P(z_1) = [0 \;\; 0]$, therefore T has to satisfy $T(z_1) = [0 \;\; 0]$.

System Identification and Model Reduction of Linear Continuous Systems from Impulse Response Data *

P. Agathoklis and V. Sreeram
Department of Electrical and Computer Engineering
University of Victoria
P.O. Box 1700, Victoria, B.C.
CANADA V8W 2Y2

Abstract

Two new algorithms for identification and model reduction are proposed. They are based on the weighted impulse response grammians introduced in this paper for stable linear continous systems. The reduction algorithm yields a family of stable reduced order models for a stable original system. The first of them matches the impulse response norm of the original system and its derivatives exactly. Finally, the method is illustrated by a numerical example and is compared with well known balanced realization techniques.

1 Introduction

The problem of approximating a system using a linear model has attracted considerable attention. In system identification one is concerned with the determination of the model parameters using measurements of the input-output data [1,2]. In model reduction the original model is given and it is approximated by a lower order model [3,4]. In both cases the objective is to obtain a model representing the original system as accurately as possible.

In this paper, two new techniques for system identification and model reduction are presented. Both the techniques are based on the *weighted impulse response grammian* which is introduced here for linear continous systems and has entries formed using the weighted performance criteria

$$I_r \triangleq \int_0^\infty t^r y^2(t)dt \quad r = 0,1,2,\ldots$$

with $y(t)$ representing the impulse response and its $n-1$ derivatives. This grammian contains direct information on the input-output behaviour of the system and is therefore, a good candidate for use in identification and model reduction. In particular it is

---

*This work has been supported by NSERC.

shown [7] that its rank is equal to the order of the system and that two grammians corresponding to consecutive values of r form a positive definite solution to the Lyapunov equation with the system matrix in controllability canonical form. This property is the basis of the two algorithms to be presented in this paper.

The new identification algorithm is developed using the weighted impulse response grammian. It is shown that the system parameters can be obtained as the solution to a sytem of linear equations with coefficients obtained from the numerical evaluation of the weighted performance criteria using the measurements of the impulse response. The numerical aspects of computing these coefficients are also briefly discussed.

The new approach to the model reduction problem is also based on the weighted impulse response grammian. Several other approaches to this problem have been already presented in the literature (see bibliography in [3], [4]). Some of them attempt to approximate the input-output behaviour of the original system by retaining the contributions of the dominant eigenvalues while the contribution of the non-dominant eigenvalues are neglected. Another group is based on obtaining a model by matching certain response of the system using optimization while other methods are based on Routh-array [8,19] and Pade' approximations. An important contribution to the model reduction problem is the introduction of system balancing by Moore [10] which describes the input-output behaviour using the second order modes. This leads to a number of techniques for model reduction [11,12,13,14,17,20,21]. In this paper the input-output behaviour is described using the impulse response grammian. This leads to a technique which guarantees that the norm of the impulse response of the reduced order model $h_k(t)$ and its $k-1$ derivatives

$$\| \overset{i}{h_k}(t) \| \;=\; \int_0^\infty (\overset{i}{h_k}(t))^2 dt$$

will match exactly the norm of the impulse response of the original system and its $k-1$ derivatives, i.e.,

$$\| \overset{i}{h}(t) \| \;=\; \| \overset{i}{h_k}(t) \| \quad i = 0, 1, \ldots, k-1$$

As expected, this method matches the impulse response better than earlier methods which only minimize the error between the impulse response norms of the original and reduced order systems. It is shown that the proposed reduced order model can match the impulse response as good as the reduced order model obtained through balanced realization techniques. At the same time the new method matches exactly the impulse response norms while the balanced realization gives a finite error.

The paper is organised as follows: In Section 2, the weighted impulse response grammian is presented and its properties are discussed. In section 3 and 4 the identification and model reduction algorithms are presented and section 5 the algorithms are illustrated by a numerical example.

2 The weighted performance criteria and the weighted impulse response grammian

Consider a SISO continuous system described by the following minimal realization

$$x(t) \;=\; Ax(t) + bu(t) \tag{1}$$

$$y(t) \;=\; cx(t) \tag{2}$$

where $x(t) \in R^n$. The weighted performance criteria are defined as

$$I_r \;=\; \int_0^\infty t^r y^2(t)dt \quad r = 0,1,2,\ldots \tag{3}$$

It is shown in [5,6,7] that this weighted cost performance criteria can be evaluated in a closed form as

$$I_r \;=\; \int_0^\infty t^r y^2(t)dt = \int_0^\infty t^r x^T(t)Q_0 x(t)dt = r!x^T(0)Q_{r+1}x(0) \tag{4}$$

$$\text{where } Q_0 \;=\; c^T c \tag{5}$$

where Q_{r+1} can be obtained recursively from the following set of Lyapunov equations

$$A^T Q_{i+1} + Q_{i+1}A \;=\; -Q_i \text{ for } i = 0,1,\ldots,r \tag{6}$$

The solution of eqn. 1 can be expressed as

$$x(t) \;=\; e^{At}x(0+) \tag{7}$$

where the initial condition x(0+) is set up using an input signal

$$u(t) \;=\; g_1\delta(t) + g_2\delta^1(t) + \ldots + g_n\delta^{n-1}(t) \tag{8}$$

with $\delta(t)$ the impulse function and $\delta^j(t)$ the j^{th} derivative of $\delta(t)$. Assuming x(0-) = 0, x(0+) is given by

$$x(0+) \;=\; \begin{bmatrix} b \; Ab \; \ldots \; A^{n-1}b \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_n \end{bmatrix} = \mathcal{C}g \tag{9}$$

where $\mathcal{C}$ is the controllability matrix of eqn. 1 and is nonsingular since {A,b,c} is a minimal realization. From eqns. 4, 7, and 9, it follows that

$$I_r = \int_0^\infty t^r \mathrm{g}^T \mathcal{C}^T e^{A^T t} c^T c e^{At} \mathcal{C} g dt \;=\; r!\mathrm{g}^T \mathcal{C}^T Q_{r+1} \mathcal{C}\mathrm{g} = r!\mathrm{g}^T \hat{Q}_{r+1}\mathrm{g} \tag{10}$$

Eqn. 10 implies

$$\hat{Q}_{r+1} \;=\; \frac{1}{r!}\int_0^\infty t^r (ce^{At}\mathcal{C})^T (ce^{At}\mathcal{C})dt \tag{11}$$

$$= \frac{1}{r!}\int_0^\infty t^r \begin{bmatrix} h(t)^2 & h(t)\dot{h}(t) & \ldots & h(t)\overset{n-1}{h}(t) \\ h(t)\dot{h}(t) & \ddots & \ldots & \dot{h}(t)\overset{n-1}{h}(t) \\ \vdots & & \vdots & \vdots \\ h(t)\overset{n-1}{h}(t) & \ldots & \ldots & \overset{n-1}{h^2}(t) \end{bmatrix} dt$$

$$\text{where } h(t) \;=\; ce^{At}b \tag{12}$$

is the impulse response of the system and $\overset{(i-1)}{h}(t)$ is the $(i-1)^{th}$ derivative of h(t). Let $\hat{Q}_{r+1}$ be called the *weighted impulse response grammian* and for r = 0, $\hat{Q}_1$, the *impulse response grammian.* $\hat{Q}_{r+1}$ can be obtained recursively from

$$\hat{A}^T\hat{Q}_{i+1} + \hat{Q}_{i+1}\hat{A} \;=\; -\hat{Q}_i \text{ for } i = 0,1,\ldots,n \tag{13}$$

$$\text{where } \hat{A} \;=\; C^{-1}AC \quad \text{and } \hat{Q}_0 = C^Tc^TcC = \hat{c}^T\hat{c} \tag{14}$$

It is obvious that $\hat{A}$ and $\hat{c}$ are in controllability canonical form [8].

To the best of our knowledge the term *impulse response grammian* has not appeared in the control literature so far. This matrix contains information about the input output behaviour of the system, and therefore, it seems to be a good candidate for model reduction like controllability and observability grammians used in Moore [10] and cross grammian used in Fernando & Nicholson [12]. This will be discussed in the section 4. The rank of the grammian $\hat{Q}_{r+1}$ for a n-th order controllable and observable system is n, as it can be easily seen from eqn. 11. Further, in the following lemma it will be shown that the *(n+1)* x *(n+1)* grammian $\hat{Q}^{n+1}_{r+1}$ has also rank n.

Lemma 1: *The weighted impulse response grammian of order (n+1)* x *(n+1) is positive semidefinite of rank n for an n-th order controllabe and observable system.*

Proof : See [7].

The above *lemma* indicates that an estimate of the sytem order n can be obtained by considering the rank of $\hat{Q}_i$. The order n is given by the highest order of the grammian $\hat{Q}_{r+1}$ that is positive definite. It should be noticed that the numerical rank determination is an ill-conditioned problem and that further numerical errors, which will be discussed in section 3, are introduced during the numerical evaluation of the eqn. 11. The effect of these errors could be reduced by considering the locations of the singular values [18] of a series of grammians, $\hat{Q}_i, \quad i = 1,2,\ldots$.

The weighted impulse response grammian derived in this section is used for identification and model reduction in the following two sections.

3 Identification

The method to be presented in this section is used to identify the sytem, eqns. 1 and 2, using measurements of the impulse response *h(t)* which are used to compute the weighted impulse response grammians $\hat{Q}_r$ and $\hat{Q}_{r+1}$ using eqn. 11. The matrix $\hat{A}$ is obtained using the fact that $\hat{A}$, $\hat{Q}_r$ and $\hat{Q}_{r+1}$ satisfy the following Lyapunov equation:

$$\hat{A}^T\hat{Q}_{r+1} + \hat{Q}_{r+1}\hat{A} \;=\; -\hat{Q}_r \tag{15}$$

where $\hat{A}$ is in controllability canonical form. Eqn. 15 is symmetric and yields $n(n+1)/2$ equations. The $n(n-1)/2$ of them are the following:

$$\hat{q}^{r+1}_{i,i+1} \;=\; \frac{1}{2}\hat{q}^{r}_{ii} \quad for\ i = 1,2,\ldots,n-1 \tag{16}$$

and

$$\hat{q}_{ij}^{r+1} \quad = \quad -\hat{q}_{i,j-1}^{r} - \hat{q}_{i+1,j-1}^{r+1} \tag{17}$$

$$i = 1, 2, \ldots, n-2;$$
$$j = 3, 4, \ldots, n;$$
$$and \quad j \geq i+2$$

Eqns. 16 and 17 involve only entries of $\hat{Q}_r$ and $\hat{Q}_{r+1}$ and can be used to simplify the computation of these matrices. Instead of computing all the entries of $\hat{Q}_{r+1}$ using eqn. 11, which involves an infinite integral, one can evaluate only the diagonal elements of $\hat{Q}_{r+1}$ using eqn. 11, and all the off-diagonal elements can be computed using the eqns. 16 and 17 in a proper order.

The remaining n equations give:

$$\begin{array}{ccccccccc}
\hat{q}_{11}^{r+1} a_1 & + & \hat{q}_{12}^{r+1} a_2 & + & \ldots & + & \hat{q}_{1n}^{r+1} a_n & = & -\hat{q}_{1n}^{r} - \hat{q}_{2n}^{r+1} \\
\hat{q}_{12}^{r+1} a_1 & + & \hat{q}_{22}^{r+1} a_2 & + & \ldots & + & \hat{q}_{2n}^{r+1} a_n & = & -\hat{q}_{2n}^{r} - \hat{q}_{3n}^{r+1} \\
\vdots & & & & \vdots & & & & \vdots \\
\hat{q}_{1n}^{r+1} a_1 & + & \hat{q}_{2n}^{r+1} a_2 & + & \ldots & + & \hat{q}_{nn}^{r+1} a_n & = & -\hat{q}_{nn}^{r}/2
\end{array}$$

which can be used for the identification of the n coefficients a_i of the characteristic polynomial. The above equation can be written as,

$$\hat{Q}_{r+1}\mathbf{a} \quad = \quad \mathbf{q} \tag{18}$$

where

$$\mathbf{a} \quad = \quad [a_1,\ a_2,\ \ldots\ , a_n]^T \tag{19}$$

$$\mathbf{q} \quad = \quad [-\hat{q}_{1n}^{r} - \hat{q}_{2n}^{r+1},\ -\hat{q}_{2n}^{r} - \hat{q}_{3n}^{r+1},\ \ldots\ , -\hat{q}_{nn}^{r}/2]^T \tag{20}$$

Eqn. 18 has a unique solution, since $\hat{Q}_{r+1}$ is always positive definite for a stable system, and $\hat{A}$ is given by

$$\hat{A} = \begin{bmatrix} 0 & \ldots & 0 & a_1 \\ 1 & \ldots & 0 & a_2 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \ldots & 1 & a_n \end{bmatrix} \tag{21}$$

The input and output vectors are given by

$$\hat{b}^T \quad = \left[\ 1,\ 0,\ \ldots, 0\ \right] \quad \hat{c} = \left[\ h(t),\ \dot{h}(t),\ \ldots,\ \overset{(n-1)}{h}(t)\ \right]_{\{t=0\}} \tag{22}$$

The proposed algorithm for identification can now be summarized as follows:

(i) Compute all the elements of $\hat{Q}_r$ using eqn. 11.

(ii) Compute all the diagonal elements of $\hat{Q}_{r+1}$ using

$$\hat{q}_{ii}^{r+1} \quad = \quad \frac{1}{r!}\int_0^\infty t^r\, \overset{i-1}{h}{}^2(t)\,dt \tag{23}$$

(iii) Compute all the off-diagonal elements of $\hat{Q}_{r+1}$ using eqn. 16 and 17 in a proper order.

(iv) Solve eqn. 18 for a and obtain $\{\hat{A}, \hat{b}, \hat{c}\}$ using eqn. 21 and 22.

The proposed algorithm gives, theoretically, the same system model for any value value of r. In step (i), however, it is required to evaluate the infinite integral eqn. 23, and this introduces errors due to numerical integration and differentiation and due to the use of a finite number of measurements. These errors can be reduced by using a low value of r so as to increase the weighting of measurements for values of t close to zero, and by carefully choosing the numerical techniques used. This will be illustrated in section 5 with an example.

4 Model Reduction

Consider a stable linear multivariable system described by eqs. 1 and 2. The objective is to obtain a reduced-order model

$$\begin{aligned} x_k(t) &= \hat{A}_k x_k(t) + \hat{b}_k u(t) &(24)\\ y_k(t) &= \hat{c}_k x_k(t) &(25)\end{aligned}$$

with $x_k \in R^k, \quad k < n$, so that $y_k(t)$ is a close approximation to $y(t)$. The method proposed in this section is based on obtaining a reduced order model such that

$$\| \overset{i}{h}_k \| = \int_0^\infty (\overset{i}{h}_k(t))^2 dt = \| \overset{i}{h} \| = \int_0^\infty (\overset{i}{h}(t))^2 dt \qquad (26)$$

$$i = 0, 1, \ldots, (k-1)$$

where $h(t)$ is the impulse response of the original system and $h_k(t)$ is the impulse response of the reduced order model. The impulse response grammian $\hat{Q}_1^n$ for the original system can be computed from

$$\hat{A}^T \hat{Q}_1^n + \hat{Q}_1^n \hat{A} = -\hat{Q}_0^n \qquad (27)$$

where $\hat{A}$ and $\hat{Q}_0^n$ are given by eqn. 14. From eqn. 11 follows immediately that the reduced order model will satisfy eqn. 25, if the diagonal elements of the impulse response grammian $\hat{Q}_1^k$ of the reduced order model are the same as the first k diagonal elements of $\hat{Q}_1^n$. Further, from eqn. 13 follows that $\hat{A}_k$, the system matrix of the reduced order model in controllability canonical form, will satisfy the following Lyapunov equation:

$$\hat{A}_k^T \hat{Q}_1^k + \hat{Q}_1^k \hat{A}_k = -\hat{Q}_0^k \qquad (28)$$

$\hat{Q}_0^k$ is chosen as the k x k leading principal minor of $\hat{Q}_0^n$ so that the reduced order model matches exactly the first k Markov parameters of the original system. The choice of $\hat{Q}_0^k$, the equality of the k diagonal elements of $\hat{Q}_1^k$ and $\hat{Q}_1^n$ and eqns. 16 and 17 imply that $\hat{Q}_1^k$ is the k x k leading principal minor of $\hat{Q}_1^n$. Hence, $\hat{A}_k$ can be obtained by solving eqn. 28 for $\hat{A}_k$ as discussed in the previous section. The proposed model reduction algorithm can now be summarized as follows:

(i) Compute the impulse response grammian $\hat{Q}_1^n$ for a given system {A,b,c} by solving the Lyapunov equation 27.

(ii) Find the impulse response grammian of the reduced order model $\hat{Q}_1^k$ and $\hat{Q}_0^k$ by taking the k x k leading principal minors of $\hat{Q}_1^n$ and $\hat{Q}_0^n$ respectively.

(iii) Compute $\hat{A}_k$, the system matrix of the reduced order model in controllability canonical form, from

$$\hat{Q}_1^k \mathbf{a_k} = \mathbf{q_k} \tag{29}$$

where

$$\mathbf{a_k} = [a_1', a_2', \ldots, a_k']^T \tag{30}$$

$$\mathbf{q_k} = [-\hat{q}_{1k}^0 - \hat{q}_{2k}^1, -\hat{q}_{2k}^0 - \hat{q}_{3k}^1, \ldots, -\hat{q}_{kk}^0/2]^T \tag{31}$$

(iv) Obtain $\hat{b}_k$ and $\hat{c}_k$ in controllability canonical form, from the eqn. 22 using the first k Markov parameters of the original system.

Clearly, the reduced order system obtained using the above technique is stable because of eqn. 28 and matches exactly the first k Markov parameters of the original system. Further, the norm of the original system impulse response and its $k-1$ derivatives are matched exactly by the reduced order model, eqn. 26. This compares well with well known model reduction techniques which minimizes the error of the impulse response norm only. Note also that $\hat{Q}_1^n$ is obtained by soving the Lyapunov equation 27 and therefore, there are no numerical errors associated with the evaluation of infinite integrals as was the case with the first step of the identification algorithm discussed in the previous section.

The proposed technique can be easily generalized to use $\hat{Q}_{r+1}$ and $\hat{Q}_r$ for any $r > 0$. This leads to the following Lyapunov equation for $\hat{A}_k$

$$\hat{A}_k^T \hat{Q}_{r+1}^k + \hat{Q}_{r+1}^k \hat{A}_k = -\hat{Q}_r^k \tag{32}$$

where $\hat{Q}_{r+1}^k$ and $\hat{Q}_r^k$ are the k x k leading principal minors of the weighted impulse response grammians of the original system $\hat{Q}_{r+1}^n$ and $\hat{Q}_r^n$ respectively. $\hat{A}_k$ can be obtained by solving eqn. 32 as discussed in section 3 and the input and output vectors $\hat{b}_k$ and $\hat{c}_k$ are chosen as in eqn. 22. The resulting reduced order model is again stable due to eqn. 32 and matches the first k Markov parameters exactly. However, it does not satisfy eqn. 26 but approximates

$$\int_0^\infty t^r (\overset{i}{h}(t))^2 dt \quad i = 0, 1, \ldots, k-1 \tag{33}$$

as can be seen from eqn. 11. It is clear that for different values of r, the method yields different models. For large values of r, it can be easily shown that the model obtained retains the first k eigenvalues of the system like the model obtained by Aggregation Method [22].

5 Numerical Example

Consider the frequently used 3rd order system [19] described by the following transfer function:

$$G(s) = \frac{8s^2 + 6s + 2}{s^3 + 4s^2 + 5s + 2}$$

whose poles are at s = -1, -1, -2.

This system will be used to illustrate the use of the identification and model reduction algorithms proposed. For identification, impulse response measurements at discrete intervals of T = 0.1 sec between 0 and 15 sec are generated. The diagonal elements of $\hat{Q}_i$, $i = 1,2$ are computed using eqn. 23. Simpson's 1/3 rule [15] is used for numerical integration and Stirling's formula [15] for numerical differentiation, and the accuracy of both these methods were increased by Romberg's technique [16]. The off-diagonal elements have been computed using eqns. 16 and 17. This yields the $\hat{Q}_i$ matrices, which are not shown here for page restrictions.

Using $\hat{Q}_0$ and $\hat{Q}_1$, the identification technique yields following transfer function

$$G(s) = \frac{8.0s^2 + 5.9968s + 2.0001}{s^3 + 3.9996s^2 + 4.9987s + 1.9988}$$

It can be easily verified that $\hat{Q}_1$ and $\hat{Q}_2$ matrices also yield the same system model. Further, the matrices $\hat{Q}_2$ and $\hat{Q}_3$, or any two successive $\hat{Q}_i$ matrices, should also give the same system model provided that the errors due to numerical integration and differentiation in eqn. 23 are negligible.

The model reduction technique proposed in section 4 is used to reduce the order of the above system. The 2 x 2 impulse response grammian and $\hat{Q}_k^0$ for the reduced order model are obtained as the 2 x 2 leading principal submatrices of $\hat{Q}_1$ and $\hat{Q}_0$. Using $\hat{Q}_1^k$ and $\hat{Q}_0^k$ and eqn. 29, 2-nd order model obtained is

$$\hat{A}_k^0 = \begin{bmatrix} 0 & -3.4336 \\ 1 & -3.5798 \end{bmatrix} \quad \hat{b}_k^0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \hat{c}_k^0 = \begin{bmatrix} 8 & -26 \end{bmatrix}$$

For comparison purposes a second order model is obtained using balanced realization techniques [10,12,13,14]. It is given by

$$A_k^{bal} = \begin{bmatrix} -3.4985 & 1.8966 \\ -1.8966 & -0.0541 \end{bmatrix} \quad b_k^{bal} = \begin{bmatrix} 2.8294 \\ 0.2804 \end{bmatrix} \quad c_k^{bal} = \begin{bmatrix} 2.8294 & -0.2804 \end{bmatrix}$$

The models are compared using the following error measures. The maximum error between the impulse responses of the original system and the reduced order model defined as

$$err = \max|h(t) - h_k(t)|$$

is 0.067778 for the model $\{\hat{A}_k^0, \hat{b}_k^0, \hat{c}_k^0\}$ and 0.073148 for $\{A_k^{bal}, b_k^{bal}, c_k^{bal}\}$. Further, consider the error in the impulse response norm defined as

$$\epsilon = \frac{\|h\|^2 - \|h_k\|^2}{\|h\|^2} \text{ x } 100$$

which is a criterion often used in the comparison of model reduction techniques [14]. For the model $\{\hat{A}_k^0, \hat{b}_k^0, \hat{c}_k^0\}$ error, $\epsilon = 0.0$, whereas for the model $\{A_k^{bal}, b_k^{bal}, c_k^{bal}\}$, $\epsilon = 0.0649$. It is clear that the balanced realization techniques [10,12,13,14], produce a finite error in the impulse response norm while the proposed technique provides a simple way of matching the impulse response norm exactly.

6 Conclusion

Two new methods for identification and model reduction of linear continuous sytems based on weighted impulse response grammians are proposed. The grammians which are introduced in this paper, represent the input-output behaviour of a system and are of rank n for n-th order system. Two grammians corresponding to successive r (order of the weighting factor) form a positive definite solution to the Lyapunov equation with system matrix in controllability form. In identification the system parameters are obtained by solving the Lyapunov equation using any two successive grammians which are computed from impulse response measurements. In model reduction, the Lyapunov equation for the reduced order system is solved using the leading principal minors of any two successive grammians of the original system. This leads to a family of reduced order models which retain stability and the first k Markov parameters of the original system. The best match of the original system impulse response is obtained by the first model ($r = 0$) which matches exactly the original system's impulse response norm and the norms of its $k - 1$ derivatives. In this aspect the proposed method gives better results than balanced realization techniques which only approximate the impulse response norm.

References

[1] P. Eykhoff. *"System Identification, parameter and state estimation"* Wiley, London, 1974.

[2] L. Ljung, *"System Identification"*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1987.

[3] N.K. Sinha, and Kuszta *"Modelling and Identification of Dynamic Systems"*, Van Nostrand Reinhold, New York, 1983.

[4] M.S. Mahmoud and M.G. Singh. *"Large Scale Modelling"*, Pergamon Press, London, 1981.

[5] A.G.J. McFarlane. *"The calculation of functionals of the time and frequency response of a linear constant coefficient dynamical system"*, Quart. J. Mech. Applied Math., vol. 16., pt. 2, pp. 259-271, 1963.

[6] M. Mansour, E.J. Jury, and L.F. Chaparro *"Estimation of the margin of stability for linear continuous and discrete systems"*, Int. J. Control, 1979, vol 30 pp. 49-69.

[7] P. Agathoklis, and V. Sreeram *"Lower bounds for the stability margin of linear Continuous systems based on the Weighted Performance Criteria"*, accepted for publication in Proc. IEE, Part-D.

[8] T. Kailath. *"Linear Systems"*, Prentice-Hall Inc. Englewood Cliffs., N.J., 1980.

[9] M.F. Hutton, and B. Friedland *"Routh Approximations for Reducing the order of Linear Time-Invariant Systems"*, IEEE Trans. on Automat. Contr. vol. AC-30, No.8, pp. 329-337, 1975.

[10] B.C. Moore. *"Pricipal component analysis in Linear Systems: Controllability, Observability, and Model Reduction"*, IEEE Trans. on Automat. Contr. AC-26, pp. 17-31, 1981.

[11] L. Pernebo and L.M. Silverman. *" Model reduction via balanced state space representations"*, IEEE Trans. on Automat. Contr. AC-27, pp. 382-387, 1982.

[12] K.V. Fernando and H. Nicholson. *"On the Structure of balanced and other principal representations of SISO systems"*, IEEE Trans. on Automat. Contr. AC-28, pp. 228-231, 1983.

[13] P.T. Kabamba *"Balanced Gains and Their Significance for L^2 Model Reduction"*, IEEE Trans. on Automat. Contr., vol., AC-30, pp. 690-693, 1985

[14] A.M. Davidson *"Balanced Systems and Model Reduction"*, Electronics Letters, vol. 22, No.10, pp. 531-532, 1986.

[15] F. Scheid. *"Theory and Problems of Numerical Analysis"*, Schaum's Outline Series, McGraw Hill Book Company, New York, 1968.

[16] C. F. Gerald. *"Applied Numerical Analysis"*, Addison-Wesley Publishing Company Inc., 1978.

[17] R.E. Skeleton, and A. Yousuff. *"Component Cost Analysis of Large Scale Systems"* Int.J.Control, vol. 37, pp. 285-304, 1983.

[18] V. Klema, and A.J. Lamb, *"The Singular Value Decomposition: Its Computation and Some Applications"*, IEEE Trans. on Automat. Contr. AC-25, pp. 164-176, 1980.

[19] T.N. Lucas, *"Linear System Reduction by Impulse Energy Approximation"*, ibid, AC-30, pp. 784-786, 1984.

[20] P. Agathoklis and V. Sreeram, *"Truncation criteria for Model Reduction using Balanced Realization Technique"*, Elect. Lett., vol. 24, pp. 837-838, 1988.

[21] V. Sreeram and P. Agathoklis, *"Model Reduction using Balanced Realizations with Improved Low Frequency Behaviour"*, accepted for publication in Systems and Control Letters.

[22] M. Aoki, *"Control of Large Scale Systems by Aggregation"*, IEEE Trans. on Automat. Contr., AC-13, pp. 246-253, 1968.

NEAR-OPTIMAL MODEL REDUCTION IN BALANCED OR MODAL COORDINATES

WODEK GAWRONSKI and JER-NAN JUANG
NASA Langley Research Center, Spacecraft Dynamics Branch,
M.S. 230, Hampton, VA 23665

The paper presents necessary and sufficient conditions for near-optimal model reduction in balanced or modal coordinates. Different forms of balanced representations are introduced (Moore, Schur, Hessenberg), and their application to model reduction is discussed.

1. INTRODUCTION

Model reduction of large-scale systems has been considered in many papers. In particular, optimal reduction is considered in [6], [14], [15]. The iterative procedures developed for the optimal reduction methods are not numerically inefficient in the sense that the convergence is not guaranteed, and the bound on number of iterations can not determined. On the other hand, the numerically efficient procedures presented in [3], [5], [10], [11], [12], are not proved to be optimal, although in many cases, they give satisfactory results. This paper explains under what conditions the near-optimal results are obtained. Balanced representation assumed in the paper has equal controllability and observability grammians, but not necessarily diagonal. The family of balanced representations is expanded including Moore, Schur and Hessenberg representations, and their applications to model reduction are investigated. Grammians and reduced models are determined using the impulse or white noise as the system input which acts continuously within the infinite time domain. The impulse, or white noise excites all system frequencies evenly, and therefore is useful for the reduction purposes. Nevertheless, in special cases, when the reduction in limited time or frequency range is preferred, the procedure from [4] may be applied. The reduced model is obtained by truncating the selected system coordinates. Such a reduction can be successful only for specific representations, such as balanced and modal representations. The paper presents conditions for the reduced model to be near the global optimum. The basic approach of this paper is to take advantage of simplicity of already known reduction methods such as [10], [11], [12] and show that the almost optimal results can be obtained in a relatively simple way.

2. REDUCTION TECHNIQUE

Let the system state vector be partitioned as follows $x^T=[x_1^T \; x_T^T]$, and its representation (A,B,C) be divided appropriately

$$A=\begin{vmatrix} A_R & A_{RT} \\ A_{TR} & A_T \end{vmatrix}, \quad B=\begin{vmatrix} B_R \\ B_T \end{vmatrix}, \quad C=[C_R \; C_T]. \qquad (2.1)$$

The reduced model is obtained by deletion of the last n-k rows of A,B, and the last n-k columns of A,C, so that $A_R=LAL^T$, $B_R=LB$, $C_R=CL^T$, $L=[I_k \; 0]$. The system before reduction can be transformed to a suitable representation through an appropriate similarity transformation. The near-optimality conditions presented further in the paper provide ways to determine the coordinates to be truncated.

Introduction, according to [15], of the auxiliary system (A_e,B_e,C_e), with $A_e=\mathrm{diag}(A, \; A_R)$, $B_e^\star=[B^\star,B_R^\star]$, $C_e=[C \; -C_R]$, helps to determine the properties of balanced and modal systems. The grammians of (A_e,B_e,C_e)

$$W_{ce}=\left|\begin{array}{c|c} W_{cO} & W_{cOR} \\ \hline W^T_{cOR} & W_{cR} \end{array}\right| = \left|\begin{array}{cc|c} W_{c1} & W_{c1T} & W_{c1R} \\ W^T_{c1T} & W_{cT} & W_{cTR} \\ \hline W^T_{c1R} & W^T_{cTR} & W_{cR} \end{array}\right|, \quad W_{oe}=\left|\begin{array}{c|c} W_{oO} & W_{oOR} \\ \hline W^T_{oOR} & W_{oR} \end{array}\right| = \left|\begin{array}{cc|c} W_{o1} & W_{o1T} & W_{o1R} \\ W^T_{o1T} & W_{oT} & W_{oTR} \\ \hline W^T_{o1R} & W^T_{oTR} & W_{oR} \end{array}\right|.$$

satisfy the following Lyapunov equations

$$A_e W_{ce} + W_{ce} A_e^\star + B_e B_e^\star = 0, \qquad W_{oe} A_e + A_e^\star W_{oe} + C_e^\star C_e = 0, \tag{2.2}$$

The reduction index $J=||y-y_R||$ is determined from

$$J^2=||y||^2+||y_R||^2-2\mathrm{tr}(C^\star C_R W_{cRO})=||y||^2+||y_R||^2-2\mathrm{tr}(BB_R^\star W_{ORO}), \tag{2.3}$$

along with the normalized reduction index δ, $\delta=||y-y_R||/||y||=J/||y||$.

The distance of the reduced model from the hypothetical optimal one is evaluated from the necessary condition of optimality. For the optimal reduced model, its output y_R is orthogonal to the error $y-y_R$, so that $S=\langle y_R, y-y_R\rangle=0$. However, since $x^T=[x_1^T \;\; x_T^T]$, $y_R=C_R x_R$, $y=Cx=C_R x_1+C_T x_T$, and $W_{cRO}=[W_{cR1} \;\; W_{cRT}]$, one obtains

$$S=\langle y_R, y\rangle-\langle y_R, y_R\rangle=\mathrm{tr}(C_R^\star C_R \Delta W_c)+\mathrm{tr}(C_T^\star C_R W_{cRT}), \tag{2.4a}$$

where $\Delta W_c=W_{cR1}-W_{cR}$, and similarly

$$S=\mathrm{tr}(B_R B_R^\star \Delta W_o)+\mathrm{tr}(B_T B_R^\star W_{oRT}), \tag{2.4b}$$

where $\Delta W_o=W_{oR1}-W_{oR}$. If ϕ is the angle between the vectors y_R and $y-y_R$, then $\varepsilon_1=|\cos\phi|$ is a measure of the distance of the reduced model from the optimal one. For a near-optimal model $\varepsilon_1<<1$ is required. The above cosine is determined from

$$\cos\phi=\langle y-y_R, y_R\rangle/(||y-y_R|| \;\; ||y_R||)=(\mathrm{tr}(C^\star C_R W_{cRO})-||y_R||^2)/(J||y_R||), \tag{2.5}$$

where $||y||^2=\mathrm{tr}(C^\star C \;\; W_c)$, $||y_R||^2=\mathrm{tr}(C_R^\star C_R W_{cR})$. As a consequence of the orthogonality condition one obtains $||y-y_R||^2=||y||^2-||y_R||^2$ for the near-optimal reduction. It indicates that the index $J_p=|\;||y||^2-||y_R||^2\;|$ is an approximation of the index J, which is used as a prediction error.

For the optimal reduced model it is necessary, according to [6], [14], [15], that $\pi+I_k=0$, where $\pi=W_{oR}^{-1}W_{oOR}^\star W_{cOR}W_{cR}^{-1}$. Define the near-optimality index $\varepsilon_2=||\pi+I_k||/(||\pi||+||I_k||)$, then for the near-optimal reduction $\varepsilon_2<<1$. Finally note that any system algebraically equivalent to the near-optimal reduced model is also near-optimal, since the reduction index is invariant under similarity transformations of the reduced model.

3. NEAR-OPTIMAL REDUCTION IN BALANCED COORDINATES

The representation (A,B,C) is balanced, if its controllability and observability grammians are equal and diagonal, see [10]. In this paper this definition is expanded: a system is balanced if its observability and controllability grammians are equal, but not necessarily diagonal. In this case the balanced representation is not unique, and is invariant under an orthogonal transformation θ. Let $\bar{x}=\theta x$, and $W_c=W_o$, then also $\bar{W}_c=\bar{W}_o$, since $\bar{W}_c=\theta^{-1}W_c\theta^{-T}=\theta^T W_o\theta$, $\bar{W}_o=\theta^T W_o\theta$. This allows one to chose different forms of the balanced representation: Moore representation, where the grammians are diagonal; lower- or upper-Schur representation, where matrix A is lower- or upper-triangular; Hessenberg representation (not unique), where matrix A is in Hessenberg form. Applications of different balanced representations to model reduction are presented further in this paper.

Condition 3.1. The following $||W_{cT}||=||W_{oT}||<<||W_{cR}||=||W_{oR}||$ is a necessary and sufficient condition for near-optimal reduction in any balanced coordinates. In this case $\varepsilon_1<<1$, $\varepsilon_2<<1$, and J is close to J_p

$$J^2=J_p^2=|\ ||y||^2-||y_R||^2|=tr(C_T^{\star}C_T W_{cT})=tr(B_T B_T^{\star}W_{oT})=-2tr(A_T W_{cT}W_{oT}).$$

Proof. Sufficiency. Note that $||W_{cT}||=||W_{oT}||=0$ implies all cross-correlations with truncated subsystem equal to zero: $W_{c1T}=W_{o1T}=0$, $W_{cTR}=W_{oTR}=0$, so that from (2.2) one obtains $W_{c1}=W_{c1R}=W_{cR}=W_{o1}=-W_{o1R}=W_{oR}$, consequently $J=\delta=0$. Since J and δ are continuous functions of W_T, therefore for sufficiently small W_T the reduced model is near-optimal, $\delta<<1$. Necessity. Assume that $||W_{cT}||=||W_{oT}||\neq 0$ but J=0. Since $x_1\neq x_R$, consequently $W_{c1R}\neq\Gamma_R^2$, $W_{cTR}\neq 0$, so that, from (2.3) $J\neq 0$, which contradicts the assumption.

Corollary 3.1. In Moore balanced representation, where $W_c=W_o=\Gamma^2$, $\Gamma=diag(\gamma_i)$, $i=1,..,n$, with γ_i in decreasing order, the condition $\sum_{i=k+1}^{n}\gamma_i^4<<\sum_{i=1}^{k}\gamma_i^4$ is necessary and sufficient for near-optimal reduction.

Corollary 3.2. In Moore balanced representation as above, the condition $\gamma_{k+1}<<\gamma_k$ is necessary and sufficient for near-optimal reduction.

Corollary 3.2 is stronger then Corollary 3.1. It is well-known from [10], but its near-optimal property has not yet been proved. Our experience shows that the deletion of the part of the system, with small J_p obtained from Condition 3.1 gives also near-optimal results. Since J_p can be determined before reduction, the near-optimality is evaluated before reduction as well. In order to discuss further this problem, denote c_i the i-th column of C, and b_i the i-th row of B and define matrix Σ

$$\Sigma=diag(\sigma_i),\qquad \sigma_i=\alpha_i\gamma_i,\quad i=1,...,n, \tag{3.1}$$

where $\alpha_i^2=c_i^{\star}c_i=b_i b_i^{\star}=-2\gamma_i^2 a_{ii}$ (this equality follows from the Lyapunov equations for the controllability and observability grammians in Moore balanced representation). Denote $\Sigma=diag(\Sigma_R,\Sigma_T)$, then in Moore balanced representation $||y||^2=tr(\Sigma^2)$, $||y_R||^2=tr(\Sigma_R^2)$, and $||y||^2-||y_R||^2=tr(\Sigma_T^2)$. For the optimal reduction, $||y-y_R||^2=tr(\Sigma_T^2)$, therefore for near-optimal reduction $||y-y_R||^2=tr(\Sigma_T^2)$. Since Σ_T is known before reduction, it can serve for an a priori determination of the reduction error. This error is called the predicted error, and denoted J_p or δ_p

$$J_p^2=tr(\Sigma_T^2),\qquad \delta_p^2=tr(\Sigma_T^2)/tr(\Sigma^2). \tag{3.2}$$

In order to obtain near-minimal reduction error, rearrange the coordinates such that the matrix Σ has its diagonal entries in decreasing order, i.e.

$\sigma_i \geq \sigma_{i+1} > 0$, $i=1,..,n-1$. The approximated value of J obtained from (3.2) is the smallest possible, since Σ_T contains the smallest diagonal entries of Σ. The above is formalized as follows:

Condition 3.2. Let the system be in Moore balanced representation, and Σ as defined above, then $||\Sigma_T|| << ||\Sigma_R||$ is a necessary and sufficient condition for near-optimal reduction.

Proof. Sufficiency. Since $||\Sigma_T||=0$ implies $\sum_{i=k+1}^{n} \alpha_i^2 \gamma_i^2 = 0$, consequently, for $i=k+1,...,n$ either $\gamma_i=0$, or $\alpha_i=0$. In Moore balanced coordinates $\alpha_i=0$ implies $\gamma_i=0$, therefore $||\Sigma_T||=0$ implies $J=\delta=0$, and small $||\Sigma_T||$ implies small J and δ, i.e. near-optimal reduction. Necessity. Assume $||\Sigma_T|| \neq 0$, but $J=0$. The first inequality implies $||\Gamma_T|| \neq 0$. According to the proof of Condition 3.1 one obtains $J \neq 0$, i.e. contradiction.

Condition 3.2 is weaker than Condition 3.1, therefore more precisely determines near-optimal model. The fact that σ-evaluation of the system components gives better results than γ-evaluation was noted in [7] and [13]. The example below shows the case where reduction with γ-ranking leads to larger error than σ-ranking.

Example 3.1. The reduction in Moore balanced representation of the system given by the following triple

$$A=\begin{vmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -5.4545 & 4.5454 & 0 & -0.0273 & 0.0227 & 0 \\ 10 & -21 & 11 & 0.05 & -0.105 & 0.055 \\ 0 & 5.5 & -6.5 & 0 & 0.0275 & -0.0325 \end{vmatrix},$$

$$B^T=[0 \quad 0 \quad 0 \quad 0.0909 \quad 0.4 \quad -0.5], \quad C=[2 \quad -2 \quad 3 \quad 0 \quad 0 \quad 0],$$

is considered. Its reduction coefficients are $\sigma_1=1.9488$, $\sigma_2=1.9305$, $\sigma_3=0.4494$, $\sigma_4=0.4369$, $\sigma_5=0.3282$, $\sigma_6=0.2932$, and the related balancing coefficients are $\gamma_1=5.6089$, $\gamma_2=5.5967$, $\gamma_3=1.0998$, $\gamma_4=1.1139$, $\gamma_5=1.3447$, $\gamma_6=1.3365$. The reduction of the system by deleting the two state variables corresponding to the smallest two balancing coefficients (γ_3 and γ_4) gives the reduction error $\delta=0.2208$, with the optimality indices $\varepsilon_1=0.7708\ 10^{-3}$, $\varepsilon_2=0.0501$, and the prediction error $\delta_p=0.2201$. The reduction by deleting the two state variables corresponding to the two smallest reduction coefficients (σ_5 and σ_6) gives the reduction error $\delta=0.1544$, and the optimality indices in this case are $\varepsilon_1=0.1267\ 10^{-3}$, $\varepsilon_2=0.0303$, and the prediction error $\delta_p=0.1545$. The indices ε_1, ε_2 and the error δ show that the latter solution is closer to the optimal one.

Conditions 3.3. Let the system be in Moore balanced representation, then one of the following: 1. $||A_{RT}||^2 << ||A_T||\ ||A_R||$, and $||A_{TR}||^2 << ||A_T||\ ||A_R||$, or 2. $||B_T|| << ||B_R||$, or 3. $||C_T|| << ||C_R||$, is a necessary and sufficient condition for near-optimal reduction.

Proof. Sufficiency. Let $||A_{RT}||=||A_{TR}||=0$, then from (2.2) $W_{c1}=W_{o1}=W_{cR}=W_{oR}$, so that $W_{cTR}=W_{oTR}=0$, in consequence $W_{cOR}^T=W_{oOR}^T=[W_{cR}^T\ 0]$, i.e. $J=\delta=0$, thus the reduction is optimal. For significantly small A_{RT} and A_{TR} the reduction is, therefore, near-optimal. Necessity. Assume $||A_{RT}|| \neq 0$, $||A_{TR}|| \neq 0$, but $J=0$;

then, from (2.2) $W_{c1R}\neq\Gamma_R^2$ and $W_{cTR}\neq0$, consequently $J\neq0$, which contradicts the above assumption. Proof for the last two conditions is similar.

Condition 3.3 shows that a block-diagonal matrix A or truncated parts of B and C being zero make the reduction optimal in Moore balanced coordinates. It gives a clue for the near-optimal reduction of systems with complex poles. For a system with complex poles, the ranking of every single coordinate based on the value of σ_i does not lead to the near-optimal results. If a complex pole and its conjugate are separated during reduction then Condition 3.3 is not satisfied. In this case, its matrix A is not diagonally dominant, but block-diagonally dominant (with 2×2 blocks), and every pair of coordinates, rather then every single coordinate must be evaluated. If absolute values of the real parts of the poles are sufficiently small, the following procedure may be applied. Note that the singular values γ_i related to every 2×2 block (two complex conjugate poles) are almost equal, i.e. for $\lambda_j=\lambda_i^*$, one obtains $\gamma_i=\gamma_j$. First arrange γ_i in Γ in decreasing order, and evaluate σ_i for each γ_i from (3.1). Next, compute

$$\sigma_i=0.5(\sigma_{2i-1}+\sigma_{2i}),\quad \sigma_{i+1}=\sigma_i,\quad i=1,..,n/2 \tag{3.3}$$

and finally, put σ_i in decreasing order.

Example 3.2. Consider the system from Example 3.1, with $C=B^T$. For this case one obtains $\Sigma=diag(0.6457,\ 0.5104,\ 0.007405,\ 0.005630,\ 0.005485,\ 0.000150)$, $\Gamma=diag(1.6269,\ 0.9982,\ 0.2911,\ 1.6270,\ 0.9980,\ 0.2912)$. The reduction of the two state variables corresponding to the two smallest σ_i, namely σ_5 and σ_6 leads to the comparatively large indices $\varepsilon_1=0.4379$, $\varepsilon_2=0.4150$ and $\delta=0.8758$. This solution cannot be considered near-optimal. The reason is that the matrix A in the balanced representation is not diagonally dominant, but block-diagonally dominant. At the next step we rank not every single coordinate, but every pair of the variables, according to the procedure given above. This leads to the rearrangement of the state variables in the above representation according to the permutation matrix [4 1 2 5 6 3]. Deleting now the last two variables, (related to σ_6 and σ_3) the optimality indices $\varepsilon_1=0.9739\ 10^{-6}$, $\varepsilon_2=0.9787\ 10^{-5}$, and $\delta=0.0090$ are obtained. The obtained solution is considered close to the optimal one. The predicted error is $\delta_p=0.008998$.

In Schur balanced representation the matrix A is in the triangular form. Lower and upper Schur balanced representations are distinguished. In the first case, with the matrix A lower-triangular, $A_{RT}=0$; this implies $x_R=x_1$, consequently $W_{c1R}=W_{cR}$, and $\Delta W_c=0$ in (2.4). Equations (2.3) and (2.4) can be simplified to

$$S=tr(C_T^*C_RW_{cRT}),\quad \text{and}\quad J^2=tr(C_T^*C_TW_{cT})=-2tr(A_TW_{cT}^2). \tag{3.4}$$

In upper Schur balanced representation the matrix A is in the upper triangular form, so that $A_{TR}=0$, and, therefore $W_{o1R}=W_{oR}$ and $\Delta W_o=0$. Consequently equations (2.3), (2.4) can be simplified to $S=tr(B_TB_R^*W_{oRT})$, $J^2=tr(B_TB_T^*W_{oT})=-2tr(A_TW_{oT}^2)$. The importance of Schur balanced representation follows from the simplicity of indices ε_1, ε_2, J, which follows from (3.4). Furthermore, since $A_{RT}=0$ (or $A_{TR}=0$), there is no pole shifting in the reduced model. For Schur balanced representation the following condition applies:

Condition 3.4. In lower Schur balanced representation, the conditions $||A_{TR}||^2<<||A_T||\ \ ||A_R||$, or $||C_T||<<||C||$, and in upper Schur representation, the conditions $||A_{RT}||^2 << ||A_T||\ ||A_R||$, or $||B_T|| << ||B||$, are necessary and sufficient for near-optimal reduction.

Proof. Sufficiency. Consider lower Schur balanced representation. From (2.2), for $||C_T||=0$, or $||A_{TR}||=0$, one obtains $J=\delta=0$, so that, for sufficiently small C_T or A_{TR} near-optimal reduction is obtained. Necessity. Let either $||A_{TR}||\neq 0$, or $||C_T||\neq 0$, and $J=0$, then from (2.2) one obtains $W_{cT}\neq 0$, $W_{oT}\neq 0$, consequently, from (3.4) $J\neq 0$, which contradicts the assumption. For upper Schur balanced representation the proof is similar.

In lower Schur representation, each zero column of C indicates unobservable part of the system, and, in upper Schur form, each zero row of B indicates uncontrollable part of the system. Define $\varepsilon_c = ||C_T||/||C||$, $\varepsilon_b = ||B_T||/||B||$, then ε_c, ε_b are indicators of near-optimality.

Consider now Hessenberg balanced representation, for which A is in upper-Hessenberg form, and divide it accordingly to (2.1), so that $A_{TR} = \begin{vmatrix} 0 & a_{k+1,k} \\ 0 & 0 \end{vmatrix}$, then from Condition 3.4 it gives

Corollary 3.3. In upper-Hessenberg balanced representation, the conditions $|a_{k+1,k}|^2 << ||A_R||\ ||A_T||$, and either $||A_{RT}||^2 << ||A_R||\ ||A_T||$, or $||B_T|| << ||B||$, are necessary and sufficient for the near-optimal reduction. Similar conclusions can be given for lower Hessenberg representation. The above corollary makes the reduction clues from [9] more specific. Namely, for near-optimal reduction, not only $a_{k+1,k}$ must be small but also its lower counterpart A_{RT}, or the corresponding part B_T of B. The following example gives the comparison of the balanced model reduction in Moore, Schur, and Hessenberg coordinates.

Example 3.3, from [9]. The system given by the following triple:

$$A = \begin{vmatrix} -.21053 & -.10526 & -.0007378 & 0 & .0706 & 0 \\ 1 & -.03537 & -.000118 & 0 & .0004 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -605.16 & -4.92 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -3906.25 & -12.5 \end{vmatrix}, \quad B = \begin{vmatrix} -7.211 \\ -.05232 \\ 0 \\ 794.7 \\ 0 \\ -448.5 \end{vmatrix},$$

$$C = \begin{vmatrix} 1 & 0 & 0 & .000334 & 0 & -.007728 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{vmatrix},$$

is reduced, from 6 to 4 state variables. The results are obtained in Moore, Schur and Hessenberg balanced coordinates. For the Moore reduced model $\delta=0.3679\ 10^{-6}$, $\varepsilon_1=0.4054\ 10^{-8}$, $\varepsilon_2=0.9990\ 10^{-9}$, for the Schur reduced model $\delta=0.006119$, $\varepsilon_1=0.007797$, $\varepsilon_2=0.02075$, and for the Hessenberg reduced model $\delta=0.04033$, $\varepsilon_1=0.4649$, $\varepsilon_2=0.03910$. Additionally, for the Schur model one obtains $\varepsilon_c=0.1083$, $\varepsilon_b=0.10103$. In the balanced Hessenberg representation

$$A_{bh} = \left|\begin{array}{cccc|cc} -0.0821 & -0.2977 & -0.0159 & -0.0585 & 0.0001 & 0.0144 \\ 0.3419 & -1.4002 & -16.091 & 7.3310 & -1.0583 & -3.3351 \\ 0.0000 & 17.892 & -0.6782 & -62.269 & 0.8062 & 2.7055 \\ 0.0000 & 0.0000 & 56.950 & -11.118 & 9.1326 & 9.1189 \\ \hline 0.0000 & 0.0000 & 0.0000 & -6.1402 & -0.4334 & -25.670 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 23.446 & -3.9540 \end{array}\right|,$$

$$B_{bh}^T = [\ 3.2471 \quad -3.7422 \quad 0.2124 \quad 0.8531 \ | -0.0383 \quad -0.2029],$$

$$C_{bh} = \left|\begin{array}{cccc|cc} -0.8505 & 0.7306 & 0.4188 & 2.1342 & -0.2511 & -0.4779 \\ -3.1337 & -2.9258 & -0.2029 & -0.8254 & 0.0358 & 0.1966 \end{array}\right|.$$

the 5,4-th entry of A_{bh}, and the norm of its counterpart are small in comparison to the other lower diagonal entries of A_{bh}. Defining $\varepsilon_h=(||A_{RT}|| \ ||A_{TR}||)/(||A_R|| \ ||A_T||)$ one finds $\varepsilon_h=0.0270$, or defining $\varepsilon_h=||B_T||/||B||$ one obtains $\varepsilon_h=0.04190$.

4. NEAR-OPTIMAL REDUCTION IN MODAL COORDINATES

Let $A_{RT}=0$, $A_{TR}=0$, for the representation (A,B,C), then $y=y_R+C_Tx_T$. The error $y-y_R$ between the original and the reduced model is solely determined from the parameters of the truncated system, $y-y_R=C_Tx_T$, and its norm is

$$J^2=tr(C_T^\star C_TW_{cT})=tr(B_TB_T^\star W_{oT})=-2tr(A_TW_{cT}W_{oT})=-2tr(A_T^\star W_{oT}W_{cT}). \qquad (4.1)$$

The reduced model is optimal if the error $y-y_R$ and y_R are orthogonal. Using $y-y_R=C_Tx_T$, and $y_R=C_Rx_R$ yields $<y-y_R,y_R>=<y_T,y_R>=tr(C_T^\star C_RW_{cRT}) =tr(B_TB_R^\star W_{oRT})$, thus $\varepsilon_1=|\cos\phi|=|<y_T,y_R>|/(||y_T|| \ ||y_R||)<<1$, and $\varepsilon_2=||W_{oR}^{-1}W_{oRT}W_{cTR}W_{cR}^{-1}||<<1$ are the necessary optimality conditions. Note that reduction in the modal coordinates. do not shift the poles.

Condition 4.1. In modal coordinates either $||C_T||<<||C_R||$ and $||W_{oT}||<<||W_{oR}||$, or $||B_T||<<||B_R||$ and $||W_{cT}||<<||W_{cR}||$, or $||W_{oT}W_{cT}||<<||W_{oR}W_{cR}||$ is necessary and sufficient condition for near-optimal reduction.
Proof. Is trivial, and follows from (4.1).

The above conditions can be fairly easily evaluated, except for highly correlated states. In this case the following condition help to determine the near-optimal solution. Denote W either as $W=W_oW_c$, or as $W=W_cW_o$, and let W be divided so that $W=\left|\begin{array}{c|c} W_R & W_{RT} \\ \hline W_{TR} & W_T \end{array}\right|$, then the following condition is derived.

Condition 4.2. The condition $||W_{RT}||^2<<||W_R|| \ ||W_T||$, $||W_{TR}||^2<<||W_R|| \ ||W_T||$, and $||W_T||<<||W_R||$, is necessary and sufficient for near-optimal reduction in modal coordinates. The error is approximately determined from $J^2=-2tr(A_T^\star W_T)$, for $W=W_oW_c$, and $J^2=-2tr(A_TW_T)$, for $W=W_cW_o$.
Proof. Sufficiency. Consider first $W=W_oW_c$. We prove that $||W_T||=0$, and $||W_{RT}||=||W_{TR}||=0$ imply $W_{cRT}=W_{oRT}=0$. If $rank(W)=rank(W_c)=rank(W_o)$, then the sufficiency is obvious. From [16] it follows that for nonsingular W_c the matrices W and W_o are congruent, while for nonsingular W_o the matrices W and W_c are congruent. Consider nonsingular W_c, then $||W_T||=||W_{TR}||=||W_{RT}||=0$ yields $||W_{oT}||=||W_{oRT}||=0$, consequently $||W_{cRT}||=0$. Similarly, for nonsingular W_o Condition 4.2 leads $||W_{cT}||=||W_{cRT}||=||W_{oRT}||=0$. In consequence, W_c and W_o are block-diagonal, and $W_T=W_{cT}W_{oT}$. In this case $J=\delta=0$, since either $W_{cT}=0$, or $W_{oT}=0$. Since δ and J are continuous functions of W, Condition 4.2 leads to the near-optimal reduction. Necessity. Assume $||W_T||\neq0$, $||W_{RT}||\neq0$, $||W_{TR}||\neq0$, and $J=0$. For $J=0$, one has $\varepsilon_1=0$, but for

$\varepsilon_1=0$ one obtains, from (2.2), $\mathrm{tr}(A_T^* W_{oTR} W_{cRT} + W_{oTR}^T W_{cRT} A_R^*)=0$. For $\varepsilon_2=0$ one has $W_{oTR}^T W_{cTR}=0$, so that $\mathrm{tr}(A_T^* W_{oTR} W_{cRT})=0$. From (4.1) $J^2=-2\mathrm{tr}(A_T^* W_{oT} W_{cT})=$ $-2\mathrm{tr}(A_T^* W_T)+2\mathrm{tr}(A_T^* W_{oRT}^T W_{cRT})$ so that $J^2=-2\mathrm{tr}(A_T^* W_T)$. Since the system is stable, and $||W_T||\neq 0$, one obtains $J\neq 0$, which contradicts the assumption. Proof for $W=W_c W_o$ is similar.

Consider now a symmetric, or an orthogonally symmetric system [1], [2], then its cross-grammian exists, and $W_{co}^2=W_c W_o$. Let the cross-grammian be divided into the retained and truncated parts respectively, then the following condition is valid.

Condition 4.3. The conditions $||W_{coRT}||^2 << ||W_{coR}||\ ||W_{coT}||$, $||W_{coTR}||^2 << ||W_{coR}||\ ||W_{coT}||$, $||W_{coT}|| << ||W_{coR}||$ are necessary and sufficient for near-optimal reduction of the symmetric systems in modal coordinates. The error is approximately determined by $J^2=-2\mathrm{tr}(A_T W_{coT}^2)$.

Proof. Note that $W=W_{co}^2$. Consequently, the conditions $||W_{coT}||=0$ and $||W_{coRT}||=||W_{coTR}||=0$ imply $W_T=0$ and $W_{RT}=W_{TR}^T=0$, i.e. optimal reduction. Small W_{coT}, W_{coRT}, and W_{coTR} thus lead to the near-optimal reduction.

The broader class of near-optimal solutions is obtained, from the next condition. Denote either $\Sigma_o^2=-2AW_c W_o=-2A^* W_o W_c$, or $\Sigma_o^2=C^* C W_c$, or $\Sigma_o^2=BB^* W_o$; in modal coordinates $A=\Lambda=\mathrm{diag}(\lambda_i)$, $i=1,...,n$. Let σ_i be the i-th diagonal element of Σ_o, and

$$\Sigma=\mathrm{diag}(\Sigma_R, \Sigma_T)=\mathrm{diag}(\sigma_i),\ i=1,...,n,\quad \sigma_i=-\zeta_i w_i,\ \sigma_i \geq \sigma_{i+1}, \tag{4.2}$$

where w_i is diagonal entry of W, and $\zeta_i=\mathrm{Re}(\lambda_i)$. For two complex-conjugate poles (λ_i, and $\lambda_j=\lambda_i^*$) one obtains $w_i=w_j$, so that $\sigma_i=\sigma_j=-0.5(\lambda_i+\lambda_i^*)w_i=-\zeta_i w_i$. With these notations a broader class of near-optimal conditions is derived.

Condition 4.4. The conditions $||W_{RT}||^2 << ||W_R||\ ||W_T||$, $||W_{TR}||^2 << ||W_R||\ ||W_T||$, and $||\Sigma_T|| << ||\Sigma_R||$, are necessary and sufficient for the near-optimal reduction in modal coordinates.

Proof. Sufficiency. The condition $||\Sigma_T||=0$ implies $\sum_{i=k+1}^{n} \zeta_i w_i=0$, (in modal coordinates A_T is diagonal) so that $w_i=0$, since, for a stable system $\lambda_i\neq 0$, $i=k+1,...,n$. However, for $||W_{RT}||=||W_{TR}||=0$ one obtains $W_T=W_{cT}W_{oT}$, and from (4.1) $J=0$, i.e. the optimal reduction. The continuity of J on σ_i gives the reduction near-optimal, for small $||\Sigma_T||$. Necessity. Assume $||\Sigma_T||\neq 0$, $||W_{RT}||\neq 0$, $||W_{TR}||\neq 0$ and $J=0$. Since $||\Sigma_T||\neq 0$ implies $||W_T||\neq 0$, the proof is the same as in Condition 4.2.

Condition 4.4 is a weaker version of Conditions 4.2 and 4.3, and shows that small modal cost Σ_T and almost-diagonal grammians are sufficient to obtain near-optimal solution. This condition is analogous to Condition 3.2. However, in modal coordinates a weak correlation between state variables is required (in balanced coordinates the state variables are uncorrelated by definition). The single condition $||\Sigma_T|| << ||\Sigma_R||$, used in [11], [12], has not been proved to give the near-optimal solution.

The procedure for choosing n-k modes for reduction follows from Condition 4.3. Let Σ be defined by (4.2), with its diagonal entries in decreasing order, $\sigma_i \geq \sigma_{i+1}$, and appropriately rearrange modal (A,B,C). Delete the last n-k rows of A and B, and the last n-k columns of A and C.

Assume the system to be divided into several subsystems. The question arises, if the index J can be evaluated from the indices of each individual subsystem J_i, and if the obtained reduction is near-optimal. In general, the answer is negative, since the index J contains not only J_i, but also the cross-correlations of the subsystems. Let the truncated part of A be in the block-diagonal form $A_T=\mathrm{diag}(A_{T1},...,A_{T\alpha})$, and B_T, C_T and grammians are divided accordingly $C_T=[C_{T1},...,C_{T\alpha}]$, $B_T^T=[B_{T1}^T,...,B_{T\alpha}^T]$, and

$$W_{cT}=\begin{vmatrix} W_{cT11}\cdots W_{cT1\alpha} \\ :::::::::::::: \\ W_{cT\alpha1}\cdots W_{cT\alpha\alpha} \end{vmatrix}, \quad W_{oT}=\begin{vmatrix} W_{oT11}\cdots W_{oT1\alpha} \\ :::::::::::::: \\ W_{oT\alpha1}\cdots W_{oT\alpha\alpha} \end{vmatrix}, \quad W_T=\begin{vmatrix} W_{T11}\cdots W_{T1\alpha} \\ :::::::::::: \\ W_{T\alpha1}\cdots W_{T\alpha\alpha} \end{vmatrix}.$$

The index J is determined from (4.1)

$$J^2=\sum_{i,j=1}^{\alpha} \mathrm{tr}(C_{Tj}^{\star}C_{Ti}W_{cTij})=\sum_{i,j}^{\alpha}\mathrm{tr}(B_{Tj}B_{Ti}^{\star}W_{oTij})=-2\sum_{i,j}^{\alpha}\mathrm{tr}(A_{Ti}W_{cTij}W_{oTij}),$$

where α is the number of subsystems deleted. If the cross-correlation terms are small, the idex J for the overall system is close to the sum of J_i for each individual subsystem. If W is diagonally dominant, the index can be determined as a sum of indices of each truncated subsystem $J^2=\sum_{i=1}^{\alpha} J_i^2$, where $J_i^2=\mathrm{tr}(C_{Ti}^{\star}C_{Ti}W_{cTii})=\mathrm{tr}(B_{Ti}B_{Ti}^{\star}W_{oTii})=$ $-2\mathrm{tr}(A_{Ti}W_{cTii}W_{oTii})$. The reduction error is close to minimal, if the subsystems with the smallest indices are truncated. If A is simple and in the modal form, so that $A=\Lambda=\mathrm{diag}(\lambda_i)$, $\lambda_i=\zeta_i+j\omega_i$, then each pole λ_i is a subsystem itself. In this case the value of each mode simplifies to $J_i^2=-(|b_i|^2|c_i|^2)/2\zeta_i$, which is the modal cost obtained by Skelton and Yousuff [12]. Choosing modes with smallest J_i to be deleted yields the reduced model close to the optimal one.

Example 4.1. Example 3.1 is re-examined for the reduction in modal coordinates. Condition 4.3 was used for determination of the near-optimal reduction. The reduced model has four state variables, and the optimality indices are $\varepsilon_1= 0.1911\ 10^{-3}$, $\varepsilon_2=0.2518\ 10^{-3}$, the normalized error $\delta=0.1538$, and its prediction $\delta_p=0.1539$. Comparing the above results with Example 3.1 yields that both solutions are close to the optimal one, and that the modal reduced model is closer to the optimal model.

Example 4.2. Example 3.2 is re-examined for the reduction in modal coordinates and the reduced model has four state variables. The obtained reduced model has the optimality indices $\varepsilon_1= 0.3683\ 10^{-4}$, $\varepsilon_2=0.2823\ 10^{-5}$, the normalized error $\delta=0.008998$, and its prediction $\delta_p=0.008986$. Comparing the results with Example 3.2 yields that both solutions are very close to the optimal one. However the balanced representation gives the results slightly closer to the optimal ones.

5. CONCLUDING REMARKS

It was shown in the paper that model reduction in the balanced or modal coordinates, under certain conditions, is near-optimal. Computer codes were developed using these conditions, which appear to be useful in many applications, including model reduction of large flexible space structures. There is only one necessary and sufficient condition of optimality. Since the near-optimal reduction is ambigious, several near-optimality conditions may appear. Their common property is that they converge to the unique optimal condition. The reason for presenting the different conditions is to illustrate their different applications, as given above. Model reduction of large flexible structures is one of many important applications. For lightly damped flexible structures the modal representation is almost balanced. In most cases balanced and modal representation give results close to the optimal ones, but in specific situations they differ. These situations have not yet been fully examined, but our computational experience shows, for example, that for closely located poles the balanced reduction gives results closer to the optimal one than the modal reduction.

REFERENCES

[1] J.A. De Abreu-Garcia and F.W. Fairman, "A note on cross-grammians for orthogonally symmetric realizations," IEEE Trans. Automat. Control, vol.AC-31, 1986, pp.866-868.

[2] K.V. Fernando, and H. Nicholson, "On the cross-grammians for symmetric MIMO systems", IEEE Trans. Circuits Syst., vol.CAS-32, 1985, pp.487-489.

[3] W. Gawronski and H.G. Natke, "Balancing linear systems," Int. J. Systems Sci., vol.17, pp.237-249, 1987.

[4] W. Gawronski and J.-N. Juang, "Balanced model reduction in limited time and frequency range," AIAA Guidance, Navigation and Control Conf., 1988, pp.275-285.

[5] W. Gawronski and H.G. Natke, "Balanced state space representation in the identification of dynamical systems", in:Application of System Identification in Engineering, Ed. H.G. Natke, Springer, Wien, 1988.

[6] D.C. Hyland and D.S. Bernstein, "The optimal projection equations for model reduction and the relationships among the methods of Wilson, Skelton and Moore," IEEE Trans. Automat. Contr., AC-30, pp.1201-1211, 1985.

[7] P.T. Kabamba, "Balanced gains and their significance for L_2 model reduction," IEEE Trans. Auromat. Control, vol. AC-30, 1985, pp.690-693.

[8] R.N. Mishra and D.A. Wilson, "A new algorithm for optimal reduction of multivariable systems." Int. J. Contr., vol.31, pp.443-466, 1980.

[9] C.P. Kwong, "Optimal chained aggregation for reduced order modelling," Int. J. Control, vol.35, pp.965-982, 1982.

[10] B.C. Moore, "Principal component analysis in linear systems: Controllability, observability, and model reduction," IEEE Trans. Automat. Contr, vol.AC-26, pp.17-32, 1981.

[11] R.E. Skelton, "Cost decomposition of linear systems with application to model reduction," Int. J. Contr., vol.32, pp.1031-1055, 1980.

[12] R.E. Skelton and A. Yousuff, "Component cost analysis of large scale systems," Int. J. Contr., vol.37, pp.285-304, 1983.

[13] R.E. Skelton and P.T. Kabamba, "Comments on balanced gains and their significance for L_2 model reduction," IEEE Trans. Automat. Control, vol. AC-31, 1986, pp.796-797.

[14] D.A. Wilson, "Optimum solution of model-reduction problem," Proc. IEE, vol.119, pp.1161-1165, 1970.

[15] D.A. Wilson, "Model reduction for multivariable systems," Int. J. Contr., vol.20, pp.57-64, 1974.

[16] S.-Y. Kung, and D.W. Lin, "A state-space formulation for optimal Hankel-norm approximations," IEEE Trans. Automat. Control, vol.AC-26, pp.942-946, 1981.

A Parametric Method For Determining The Number of Signals

Qiang Wu
Daniel R. Fuhrmann

Electronic Systems and Signals Research Laboratory
Department of Electrical Engineering
Washington University
St. Louis MO 63130

Abstract

A new and more accurate method is developed to determine the number of signals in the multi-sensor direction finding problem. The information-theoretic criteria of Yin and Krishnaiah are applied to a set of new quantities which are evaluated from the log-likelihood function. These quantities have the properties required by the criteria. Since the information-theoretic criteria use these new quantities, instead of the eigenvalues of the estimated correlation matrix, this approach possesses the advantages of not requiring a subjective threshold, and also provides higher performance than when eigenvalues are used. Simulation results are presented with comparisons to those obtained from a nonparametric method given by Wax and Kailath.

1. Introduction

In the area of passive underwater and radar signal processing, several high resolution estimators for direction-of-arrival have been developed in recent years. These include the MUSIC method by Schmidt [1], the ESPRIT method by Roy, Paulraj, and Kailath [7], and the MLE method with EM algorithm by Miller and Fuhrmann [2]. The performance of all of these high resolution methods depends on the successful determination of the number of signals. As it is an important issue in signal processing, several methods for determining this parameter have been proposed.

A number of authors, e.g. Bartlett [8], Liggett [9], and Kumaresan and Tufts [10], approach this problem in a conventional way by using hypothesis testing. In hypothesis testing, a threshold determined by some subjective judgment is required, which may cause some serious inherent defects in the estimation procedure. Recently, Wax and Kailath [3], and Zhao, Krishnaiah, and Bai [5], investigated the problem by using the information-theoretic criteria introduced by Akaike (AIC) [6], Schwartz and Rissanen (MDL) [11], [12], and Zhao, Krishnaiah, and Bai (EDC) [5]. Based on these results, some flexible and generalized criteria have been contributed by Yin and Krishnaiah [4] in which an essential feature, the freedom of choosing the first term of these information-theoretic criteria, is proposed. All of these methods rely on the eigenvalues of the estimated correlation matrix; they are all nonparametric methods and their performance is bounded by the accuracy of the estimated eigenvalues.

In this paper, we present a new approach to this problem. In the proposed method, a new set of quantities, called *differential residues*, are defined using the log-likelihood functions with different hypothesized number of signals. Then these differential residues, instead of the estimated eigenvalues, are used as the variables of the r-regular functions in the generalized criteria in [4]. The estimated number of signals is determined as the value for which the chosen criterion is minimized.

2. Problem Statement And Its Likelihood Function

In array signal processing, a signal model with well-known wide applicability is described by the equation,

$$\mathbf{y}(t) = \mathbf{D}(\Theta)\mathbf{s}(t) + \mathbf{n}(t) \tag{2.1}$$

where $\mathbf{y}(t)$ is a pX1 complex observation vector, $\mathbf{s}(t)$ is a qX1 complex signal vector, $\mathbf{n}(t)$ is a pX1 complex additive noise vector, and $\mathbf{D}(\Theta)$ is a pXq matrix in which its ith column is parameterized by an unknown bounded parameter of the ith signal and any l columns having different parameters are independent as long as $l \leq p$.

We assume that the q signals are unknown except that the limit, $\lim_{N \to \infty} \frac{1}{N}\sum_{i=1}^{N} \mathbf{s}(i)\mathbf{s}^{\dagger}(i)$, exists and that $q \leq p/2$. Here the term "unknown signals" means that there is no prior information or statistics available about the signals. The noise vector $\mathbf{n}(\cdot)$ is assumed to be a multivariable Gaussian complex random vector with mean $\mathbf{0}$ and covariance $\sigma^2\mathbf{I}$, where σ^2 is an unknown scalar constant and $\mathbf{I}$ is the identity matrix.

With the above given model the problem to be solved is to determine the number q of signals from a finite set of observations $\mathbf{y}(1), \mathbf{y}(2), \cdots, \mathbf{y}(N)$.

In the method to be proposed the log-likelihood function plays an important role. For an assumed signal number k, the log-likelihood function of $\mathbf{y}(1), \mathbf{y}(2), \cdots, \mathbf{y}(N)$ can be written, with constants removed, as

$$\mathbf{L}(\Theta, \mathbf{S}) = -\frac{1}{\mathbf{N}}\sum_{i=1}^{N} |\mathbf{y}(i) - \mathbf{D}(\Theta)\mathbf{s}(i)|^2 \tag{2.2}$$

where $\Theta = [\theta_1, \theta_2, \cdots, \theta_k]^T$ and $\mathbf{S} = [\mathbf{s}(1), \mathbf{s}(2), \cdots, \mathbf{s}(N)]^T$.

After straightforward manipulations the maximum-likelihood solution of eqn. (2.2) can be obtained as [2]

$$\hat{\Theta} = \underset{\Theta}{argmax}\ -\frac{1}{N}\sum_{i=1}^{N} \mathbf{y}(i)^{\dagger}[\,\mathbf{I} - \mathbf{D}(\Theta)(\mathbf{D}^{\dagger}(\Theta)\mathbf{D}(\Theta))^{-1}\mathbf{D}^{\dagger}(\Theta)]\mathbf{y}(i) \tag{2.3}$$

$$= \underset{\Theta}{argmax}\ -tr[\,\mathbf{I} - \mathbf{D}(\Theta)(\mathbf{D}^{\dagger}(\Theta)\mathbf{D}(\Theta))^{-1}\mathbf{D}^{\dagger}(\Theta)]\hat{\mathbf{R}}_N \tag{2.4}$$

$$\hat{\mathbf{s}}(i) = (\mathbf{D}^{\dagger}(\hat{\Theta})\mathbf{D}(\hat{\Theta}))^{-1}\mathbf{D}(\hat{\Theta})\mathbf{y}(i), \tag{2.5}$$

where $\hat{\mathbf{R}}_N = \frac{1}{N}\sum_{i=1}^{N} \mathbf{y}(\mathrm{i})\mathbf{y}^{\dagger}(i)$, and

$$\max_{\Theta, \mathbf{S}} \mathbf{L}(\Theta, \mathbf{S}) = -\frac{1}{N}\sum_{i=1}^{N} \mathbf{y}(i)^{\dagger}[\,\mathbf{I} - \mathbf{D}(\hat{\Theta})(\mathbf{D}^{\dagger}(\hat{\Theta})\mathbf{D}(\hat{\Theta}))^{-1}\mathbf{D}^{\dagger}(\hat{\Theta})]\mathbf{y}(i) \tag{2.6}$$

Define the estimated signal subspace as the subspace spanned by the matrix $\mathbf{D}(\hat{\Theta})$ and the estimated noise subspace as the biggest subspace orthogonal to the estimated signal subspace. The geometric explanation of eqn. (2.6) is that each term is the squared norm of the projection of an observation $\mathbf{y}(i)$ onto the estimated noise subspace. For simplicity, with given k we define this projection projector by

$$\mathbf{P}_k(\hat{\Theta}) = \mathbf{I} - \mathbf{D}(\hat{\Theta})(\mathbf{D}^\dagger(\hat{\Theta})\mathbf{D}(\hat{\Theta}))^{-1}\mathbf{D}^\dagger(\hat{\Theta}) \tag{2.7}$$

3. An Asymptotic Property of The MLE

The consistency of the method to be proposed is based on an asymptotic property, provided in this section, of the MLE given in eqn. (2.4).

Observe eqn. (2.6). Since $\hat{\mathbf{R}}_N$ converges to the true correlation matrix $\mathbf{R}$ with probability 1, then for a given k ($k \leq p/2$), it can be shown that as $N \rightarrow \infty$ eqn. (2.6) becomes

$$\lim_{N \rightarrow \infty} \max_{\Theta} \frac{-1}{N} \sum_{i=1}^{N} \mathbf{y}^\dagger(i)\mathbf{P}_k(\Theta)\mathbf{y}(i) \tag{3.1}$$

$$\underset{a.s.}{=} \max_{\Theta} \; -tr\, \mathbf{P}_k(\Theta)\mathbf{R} \,, \tag{3.2}$$

where "$\underset{a.s.}{=}$" indicates that the equality holds with probability 1 or almost surely.

Using eqn. (3.1), (3.2) we can examine the asymptotic behavior of $\hat{\Theta}_N$. Since for $k > q$, $\hat{\Theta}_N$ has $k-q$ nuisance parameters, we cannot expect that $\hat{\Theta}_N$ will converge to a unique point with probability 1. However we can prove that $\hat{\Theta}_N$ has the following property.

Theorem 1: For $q \leq k \leq p/2$, with probability 1, $\hat{\Theta}_N$ converges to Θ in the sense that

$$\lim_{N \rightarrow \infty} \min_{\hat{\theta}_{N1}, \cdots, \hat{\theta}_{Nk}} \sum_{j=1}^{q} |\hat{\theta}_{Ni_j} - \theta_j| = 0 \,. \tag{3.3}$$

Furthermore the maximum value of the log-likelihood function converges to $-(p-k)\sigma^2$, as $N \rightarrow \infty$.

Proof: If $\hat{\Theta}_N$ does not converges to Θ, the true parameter vector, in the sense given by eqn. (3.3), then by noting that all $\hat{\Theta}_N$ are bounded, we can find a subsequence which converges to a point $\hat{\Theta}$ in normal sense. For this $\hat{\Theta}$, in the true parameter vector Θ there must exist $n (1 \leq n \leq q)$ true parameters, say, $\theta_1, \cdots$, and θ_n, which are not equal to any element of the $\hat{\Theta}$. We are going to show that the $\hat{\Theta}$ is not a maximizer of eqn. (3.2), which contradicts the procedure by which the $\hat{\Theta}$ is obtained. According to the estimated signal subspace and the estimated noise subspace, the real signal vector $\mathbf{D}(\Theta)\mathbf{s}(i)$ and the noise vector $\mathbf{n}(i)$ in the ith observation $\mathbf{y}(i)$, can be decomposed as

$$\mathbf{D}(\Theta)\mathbf{s}(i) = \mathbf{D}(\hat{\Theta})\mathbf{r}(i) + \mathbf{U}\mathbf{v}(i) = \left[\mathbf{d}(\hat{\theta}_1), \cdots, \mathbf{d}(\hat{\theta}_k)\right]\begin{bmatrix} r_{1i} \\ r_{2i} \\ \cdot \\ \cdot \\ \cdot \\ r_{ki} \end{bmatrix} + \left[\mathbf{u}_1, \cdots, \mathbf{u}_{p-k}\right]\begin{bmatrix} v_{1i} \\ v_{2i} \\ \cdot \\ \cdot \\ \cdot \\ v_{(p-k)i} \end{bmatrix} \tag{3.4}$$

and

$$\mathbf{n}(i) = \mathbf{D}(\hat{\Theta})\mathbf{t}(i) + \mathbf{U}\mathbf{w}(i) = \left[\mathbf{d}(\hat{\theta}_1), \cdots, \mathbf{d}(\hat{\theta}_k)\right]\begin{bmatrix} t_{1i} \\ t_{2i} \\ \cdot \\ \cdot \\ \cdot \\ t_{ki} \end{bmatrix} + \left[\mathbf{u}_1, \cdots, \mathbf{u}_{p-k}\right]\begin{bmatrix} w_{1i} \\ w_{2i} \\ \cdot \\ \cdot \\ \cdot \\ w_{(p-k)i} \end{bmatrix}, \tag{3.5}$$

respectively, where $\mathbf{d}(\hat{\theta}i)$ is a column of $\mathbf{D}(\hat{\Theta})$, and $\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_{p-k}$ form a standard orthogonal basis of the estimated noise subspace. Note that since $k \leq p/2$, $\mathbf{v}(i)$ can never be a null vector as long as $\mathbf{s}(i)$ is not a null vector.

Substituting eqn. (3.4),(3.5) into the log-likelihood function in eqn. (3.1), we have

$$\lim_{N\to\infty} \frac{-1}{N}\sum_{i=1}^{N} \mathbf{y}(i)^{\dagger}\mathbf{P}_k(\hat{\Theta})\mathbf{y}(i)$$

$$= \lim_{N\to\infty} -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{p-k}|w_{ji}|^2 - \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{p-k}|v_{ji}|^2 - \sum_{j=1}^{p-k}\frac{1}{N}\sum_{i=1}^{N} 2\mathrm{Re}(v_{ji}w_{ji})$$

By the assumption of Gaussian noise and as $N\to\infty$, we obtain the value of eqn. (3.2) with probability 1,

$$-tr\ \mathbf{P}_k(\hat{\Theta})\mathbf{R} \underset{a.s.}{=} -(p-k)\sigma^2 - C\ , \tag{3.6}$$

where $C = \lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{p-k}|v_{ji}|^2$. In the appendix, it is shown that $C > 0$ with probability 1.

Following the same procedure given above, we can also prove that for a sequence $\hat{\Theta}_N$ having the mentioned property the value of of eqn. (3.2) is

$$\max_{\Theta} - tr\ \mathbf{P}_k(\Theta)\mathbf{R} \underset{a.s.}{=} -(p-k)\sigma^2 \tag{3.7}$$

Thus we prove that if a sequence $\hat{\Theta}_N$ does not have the mentioned property, then there is a subsequence which converges to a point $\hat{\Theta}$ which is not a maximizer of eqn. (3.2). So we prove that Θ_N converges to the true Θ in the given sense and therefore eqn. (3.2) reaches the maximum value $-(p-k)\sigma^2$ with probability 1.

4. Estimating The Number Of Signals

We now consider the procedure for estimating the number of signals. A set of new quantities e_k, called *differential residues*, for each given k, is defined as

$$e_k = \begin{cases} \dfrac{1}{N}\displaystyle\sum_{i=1}^{N} \mathbf{y}(i)^{\dagger}\left[\mathbf{P}_k(\hat{\Theta}^k) - \mathbf{P}_{k+1}(\hat{\Theta}^{k+1})\right]\mathbf{y}(i) & k < \lfloor p/2 \rfloor \\ \dfrac{1}{N(p-k)}\displaystyle\sum_{i=1}^{N} \mathbf{y}(i)^{\dagger}\mathbf{P}_k(\hat{\Theta}^k)\mathbf{y}(i) & k = \lfloor p/2 \rfloor \end{cases} \tag{4.1}$$

where $\hat{\Theta}^k$ is the vector consisting of the k estimated parameters.

Using the theorem and eqn.(3.7) from last section, it is easy to see that when $q \leq k \leq p/2$ and $N\to\infty$,

$$e_k \underset{a.s.}{=} \sigma^2 \tag{4.2}$$

where σ^2 is the variance of the noise.

For $k<q$, the dimension of the estimated signal subspace is less than that of the real signal subspace. By noting the geometric explanation of eqn. (2.6) and the procedure of the proof of the theorem, we find that for this case, there exists some energy of signals projected onto the estimated noise subspace, which implies for $k<q$ and $N \rightarrow \infty$,

$$e_k \underset{a.s.}{>} \sigma^2 \tag{4.3}$$

Furthermore, because according to eqn. (2.6), for $k<q$, the signal energy projected onto the estimated noise subspace always has the minimum value, then it is not hard to prove that for $k<q$, the smaller the k, the larger the value of e_k. Therefore, it is concluded that as $N \rightarrow \infty$, we have

$$e_0 > e_1 > \cdots > e_{q-1} > e_q = e_{q+1} = \cdots = e_{\lfloor p/2 \rfloor} \qquad a.s. \tag{4.4}$$

Considering Theorem 1 given by Yin and Krishnaiah [4], it shows that the differential residues, $e_0, e_1, \cdots, e_{\lfloor p/2 \rfloor}$, are qualified to be the variables required by the r-function in the theorem. In order to use Yin and Krishnaiah's theorem, we have to find the convergence rate of the differential residues.

Looking at eqn. (4.1), we see that the differential residues $e_0, e_1, \cdots, e_{\lfloor p/2 \rfloor}$, are functions of $\hat{\Theta}_N$ and $\hat{\mathbf{R}}_N$, expressed as

$$e_k = \mathbf{h}(\hat{\Theta}_N, \hat{\mathbf{R}}_N) \tag{4.5}$$

Since generally the function $\mathbf{h}$ also has the first partial derivative matrix with non-zero entries with respect to $\hat{\mathbf{R}}_N$ and the convergence rate of $\hat{\mathbf{R}}_N$ is $O\left((\frac{\log\log N}{N})^{1/2}\right)$ in the almost sure sense [5], then for any given $\hat{\Theta}_N$, in term of the Taylor series, the differential residues have a convergence rate proportional to $O\left((\frac{\log\log N}{N})^{1/2}\right)$. Noting that all $\hat{\Theta}_N$ are bounded, it is easy to show that the convergence rate of the differential residues is almost surely less than or equal to $O\left((\frac{\log\log N}{N})^{1/2}\right)$. Thus, by knowing the convergence rate of e_k's, a factor C_N of the penalty term in the general criteria given in [4] can be properly chosen as $\frac{c \log N}{N}$ [4], where c is a constant.

5. Simulation Results

To verify the proposed method and illustrate its performance, simulation examples are presented in this section. Three criteria are employed here.

Criterion 1 :

$$\mathbf{Q}(k,\ C_N) = -\sum_{i=k}^{r-1} \log e_i + (r-k)\log\left(\frac{1}{r-k}\sum_{i=k}^{r-1} e_i\right) + kC_N \tag{5.1}$$

Criterion 2 :

$$\mathbf{Q}(k,\ C_N)= \frac{1}{r-k}\sum_{i=k}^{r-1}(e_i - \frac{1}{r-k}\sum_{i=k}^{r-1} e_i)^4 + kC_N \tag{5.2}$$

Criterion 3 :

$$\mathbf{Q}(k,\ C_N)= e^{(e_k - \frac{1}{r-k}\sum_{i=k}^{r-1} e_i)} + kC_N \tag{5.3}$$

where, let $r = \lceil p/2 \rceil$ and $C_N = \frac{2\log N}{N}$.

Criterion 1 has the almost same function form as that used by Wax and Kailath [3]. The second one is given by Yin and Krishnaiah [4]. The last one is our suggestion, which has the advantage that it is more sensitive to the difference between the differential residues. The consistency of all these criteria can be found by using the properties of e_k's and following Theorem 1 in [4].

The sensor array for our simulation consists of a linear array with 9 sensors (p=9) with a half-wavelength separation between sensors. The signals are two narrow band AM plane wave signals with same power and impinging from directions $\phi_1 = 49^o$ and $\phi_2 = 53^o$ respectively. Then the observation vector $\mathbf{y}(t)$ received at array is given by

$$\mathbf{y}(t) = \mathbf{D}(\Theta)\mathbf{s}(t) + \mathbf{n}(t)$$

where

$$\mathbf{D}(\Theta) = \Big[\mathbf{d}(\theta_1),\ \mathbf{d}(\theta_2)\Big],\quad \mathbf{d}(\theta_i) = \Big[\ 1,\ e^{-j\theta_i},\ \cdots,\ e^{-j(p-1)\theta_i}\Big]^T,$$

with

$$\theta_i = \pi\cos\ \phi_i\ ,$$

$$\mathbf{s}(t) = \begin{bmatrix} s_1(t) \\ s_2(t) \end{bmatrix} = \begin{bmatrix} [1+0.5\cos(0.05t+0^o)]e^{j(0.1t+90^0)} \\ [1+0.3\cos(0.08t+45^o)]e^{j(0.1t+0^0)} \end{bmatrix},$$

and the number of samples is chosen as

$$N = 20$$

The signal to noise ratio is defined as

$$SNR_i = 10\ \log\frac{E_{S_i}}{\sigma^2}$$

where E_{Si} is the power of the ith signal.

We choose the SNR of the signals as 3 db and 5 db, respectively, and used the EM algorithm [2] to calculate the 5 differential residues ($k = 0,1,2,3,4$) that are substituted into the three criteria given above. The simulation is repeated for 5 trials with two different SNR levels. The results are shown in the following tables.

e_k	Trial1	Trial2	Trial3	Trial4	Trial5
0	118.3	111.1	109.4	113.9	109.7
1	3.671	4.727	3.794	2.735	6.591
2	1.852	1.744	1.553	2.359	1.222
3	1.525	2.250	2.218	2.204	2.421
4	1.908	2.002	2.148	2.005	1.859

Table 1 Differential Residues of Each Trial (SNR = 3 db)

e_k	Trial1	Trial2	Trial3	Trial4	Trial5
0	184.6	175.6	173.6	179.1	173.8
1	4.352	5.696	4.695	3.470	7.801
2	1.743	1.682	1.488	2.268	1.156
3	1.438	2.130	2.113	2.103	2.330
4	1.960	2.047	2.194	2.060	1.905

Table 2 Differential Residues of Each Trial (SNR = 5 db)

	SNR = 3 db			SNR = 5 db		
Trials	$\hat{q}$ by C.1	$\hat{q}$ by C.2	$\hat{q}$ by C.3	$\hat{q}$ by C.1	$\hat{q}$ C.2	$\hat{q}$ by C.3
1	2	2	2	2	2	2
2	2	2	2	2	2	2
3	2	2	2	2	2	2
4	1	1	2	1	1	2
5	2	2	2	2	2	2

Table 3 Results From Differential Resides (q = 2)

λ_k	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5
1	186.7	177.5	177.1	181.2	177.5
2	5.252	5.475	4.595	4.491	5.464
3	2.730	3.527	3.338	3.698	3.447
4	1.937	3.134	2.256	2.494	2.503
5	1.800	2.025	1.941	1.968	1.988
6	1.396	1.466	1.665	1.494	1.688
7	1.069	1.090	1.057	0.899	0.919
8	0.592	0.645	0.565	0.674	0.752
9	0.438	0.411	0.351	0.297	0.337

Table 4 Eigenvalues of Each Trial (SNR = 5 db)

Trials	1	2	3	4	5
$\hat{q}$	1	1	1	1	1

Table 5 Wax & Kailath's Results (q = 2)

From the simulation results, we see that by using the differential residues criteria 1 and 2 give successful estimates of q except for trial 4. Criterion 3 gives correct estimates for all 5 trials in the both cases. Wax & Kailath's method for the same data (SNR=5 db) yields incorrect estimates of q in all 5 trials.

VI. Conclusion

We have presented a new consistent method for estimating the number of signals in the multiple-sensor direction-finding problem. Since this method is a parametric method by which the differential residues are obtained through choosing k independent parameters (the signal parameters Θ) instead of $k(2p-k)+1$ parameters (the entries in the sample correlation matrix) [4], the differential residues have a faster convergence rate than that of the eigenvalues of the estimated correlation matrix and therefore its performance is better than that of the method based on the estimated eigenvalues. The new method is computationally more complex than previous methods; this may or may not be an important consideration depending on the implementation.

Appendix

Proposition: In eqn. (3.6), $C > 0$ with probability 1.

Proof: From eqn. (3.3), we have

$$\mathbf{U}\mathbf{v}(i) = \mathbf{D}(\Theta)\mathbf{s}(i) - \mathbf{D}(\hat{\Theta})\mathbf{r}(i). \tag{A.1}$$

It follows that

$$\mathbf{v}(i) = \mathbf{U}^{\dagger}\mathbf{D}(\Theta)\mathbf{s}(i). \tag{A.2}$$

Substituting this result into the expression for C, we see that

$$C = \lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N} \mathbf{v}^{\dagger}(i)\mathbf{v}(i)$$

$$= \lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N} \mathbf{s}^{\dagger}(i)\mathbf{D}^{\dagger}(\Theta)\mathbf{U}\mathbf{U}^{\dagger}\mathbf{D}(\Theta)\mathbf{s}(i) \tag{A.3}$$

Since it is assumed that the first n ($1 < n \leq q$) elements of Θ are not equal to any elements of $\hat{\Theta}$, we may use $\mathbf{s}_n(i)$ and $\mathbf{D}_n(\Theta_n)$ to stand for the signal vector and the manifold matrix corresponding to these n elements in Θ, respectively. Then, the C is equal to

$$C = \lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N} \mathbf{s}_n^{\dagger}\mathbf{D}_n^{\dagger}(\Theta_n)\mathbf{U}\mathbf{U}^{\dagger}\mathbf{D}(\Theta_n)\mathbf{s}_n(i) \tag{A.4}$$

By the assumption that

$$\lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N}\mathbf{s}(i)\mathbf{s}^{\dagger}(i) \tag{A.5}$$

exists, we can write out

$$\lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N}\mathbf{s}_n(i)\mathbf{s}_n^\dagger(i) \underset{a.s.}{=} \mathbf{A}_n \tag{A.6}$$

where $\mathbf{A}_n$ is a semi-definite Hermite matrix with rank m $(1 < m \le n)$.

We can find an $n \times n$ standard orthogonal matrix $\mathbf{T}$ by which there exists

$$\lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N} \mathbf{T}\mathbf{s}_n(i)\mathbf{s}_n^\dagger(i)\mathbf{T}^\dagger \underset{a.s.}{=} \begin{bmatrix} \Lambda_m & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{A.7}$$

where Λ_m is a $m \times m$ diagonal submatrix with non-zero diagonal elements.

Defining

$$\mathbf{x}(i) = \mathbf{T}\mathbf{s}_n(i), \tag{A.9}$$

in terms of eqn. (A.7) we obtain

$$\lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N} x_j^\dagger(i)x_l(i) \underset{a.s.}{=} \begin{cases} \sigma_j > 0 & j = l \le m \\ 0 & j \ne l \text{ or } j = l > m \end{cases} \tag{A.9}$$

Substituting eqn. (A.8) into eqn. (A.4) yields

$$C = \lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}^\dagger(i)\mathbf{T}\mathbf{D}_n^\dagger(\Theta_n)\mathbf{U}\mathbf{U}^\dagger\mathbf{D}(\Theta_n)\mathbf{T}^\dagger\mathbf{x}(i) \tag{A.10}$$

$$= \lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}^\dagger(i)\mathbf{H}^\dagger\mathbf{U}\mathbf{U}^\dagger\mathbf{H}\mathbf{x}(i) \tag{A.11}$$

where

$$\mathbf{H} = [\mathbf{h}_1, \cdots, \mathbf{h}_n] = \mathbf{D}_n(\Theta_n)\mathbf{T}^\dagger \tag{A.12}$$

Eqn. (A.11) can be expanded as

$$\begin{aligned} C = {} & \lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{m} \mathbf{h}_j^\dagger\mathbf{U}\mathbf{U}^\dagger\mathbf{h}_j x_j^\dagger(i)x_j(i) + \frac{1}{N}\sum_{i=1}^{N}\sum_{j=m+1}^{n} \mathbf{h}_j^\dagger\mathbf{U}\mathbf{U}^\dagger\mathbf{h}_j x_j^\dagger(i)x_j(i) \\ & + \frac{1}{N}\sum_{i=1}^{N}\sum_{j\ne l}^{n} \mathbf{h}_j^\dagger\mathbf{U}\mathbf{U}^\dagger\mathbf{h}_l x_j^\dagger(i)x_l(i) \\ = {} & \sum_{j=1}^{m}\mathbf{h}_j^\dagger\mathbf{U}\mathbf{U}^\dagger\mathbf{h}_j \lim_{N\to\infty} \sum_{i=1}^{N} x_j^\dagger(i)x_j(i) + \sum_{j=m+1}^{n}\mathbf{h}_j^\dagger\mathbf{U}\mathbf{U}^\dagger\mathbf{h}_j \lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N} x_j^\dagger(i)x_j(i) \\ & + \sum_{j\ne l}^{n}\mathbf{h}_j^\dagger\mathbf{U}\mathbf{U}^\dagger\mathbf{h}_l \lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N} x_j^\dagger(i)x_l(i) \end{aligned} \tag{A.13}$$

Since $\mathbf{h}_i$ $(i = 1, \cdots, n)$ are spanned by $\mathbf{D}_n(\Theta_n)$ and $n + k \le p$, we have

$$\mathbf{h}_i^\dagger\mathbf{U}\mathbf{U}^\dagger\mathbf{h}_i > 0 \qquad i = 1, \cdots, n \tag{A.14}$$

Then, from eqn. (A.8) we can see that as $N \to \infty$, the first term of C is larger than zero and the other terms are zero, with probability 1. This completes the proof.

References

[1] R. Schmidt, " A signal subspace approach to multiple emitter location and spectral estimation", *Ph.D. Dissertation* of Stanford University, Palo Alto, CA., Nov. 1981.

[2] M. Miller and D. Fuhrmann, " Maximum likelihood estimation of directions of arrivals of narrowband signals in noise", *ESSRL Technical Report* , Dept. of Electrical Engineering, Washington Univ., St. Louis, MO., Dec. 1987.

[3] M. Wax and T. Kailath, " Detection of signals by information theoretic criteria", *IEEE Trans. Acoustics, Speech and Signal Processing* , Vol. ASSP-33, pp. 387-392, Apr. 1985.

[4] Y. Yin and P. Krishnaiah, " On some nonparametric methods for detection of the number of signals", *IEEE Trans. Acoustics, Speech and Signal Processing* , Vol. ASSP-35, pp. 1533- 1538, Nov. 1985

[5] L. C. Zhao, P. R. Krishnaiah, and Z. D. Bai, " On detection of numbers of signals in presence of white noise", *J. Multivariate Analysis.*, Vol. 20, pp. 1-25, 1986

[6] H. Akaike, " A new look at the statistical model identification ", *IEEE Trans. Automatic Control,* Vol. AC-19, NO. 6, pp. 716-723, Dec. 1974.

[7] R. Roy, A. Paulraj, and T. Kailath, " ESPRIT-a subspace rotation approach to estimation of parameters of cisoids in noise", *IEEE Trans. Acoustics, Speech and Signal Processing* , Vol. ASSP-34, pp. 1340-1342, Oct. 1986.

[8] M. S. Bartlett, " A note on the multiplying factors for various χ^2 approximations", *J. Royal Statistics Soc., ser. B,* Vol. 16, pp. 296-298, 1954.

[9] W. S. Liggett, " Passive sonar: fitting models to multiple time series", *Signal Processing,* J. W. Griffith et al., Eds. New York: Academic, 1980, pp. 135-169.

[10] R. Kumaresan and D. W. Tufts, "Estimation of arrival of multiple plane waves", *IEEE Trans. Aerospace Electronic Systems,* Vol. AES-19, pp. 123-133, 1983.

[11] G. Schwarz, " Estimating the dimension of a model", *Ann. Statistics,* Vol. 6, pp. 461-464, 1978.

[12] J. Rissanen, " Modeling by shortest data description", *Automatica,* Vol. 14, pp. 465-471, 1978.

AN EXTENDED OBE ALGORITHM FOR ARMA PARAMETER ESTIMATION

ASHOK K. RAO*, YIH-FANG HUANG* and SOURA DASGUPTA**

*Department of Electrical & Computer Engineering, University of Notre Dame, Notre Dame, IN 46556. ** Department of Electrical & Computer Engineering, University of Iowa, Iowa City, Iowa 52242.

ABSTRACT

Recently, there appears to be a resurgence of interest in the area of membership-set parameter estimation. One such algorithm, the so-called Optimal Bounding Ellipsoid (OBE) algorithm, proves to be appealing in theory and practice. The algorithm, which features a discerning update strategy, was first derived for the recursive estimation of parameters of autoregressive with auxiliary input (ARX) models. This paper investigates an extension of the algorithm to ARMA parameter estimation. The convergence analysis is complicated due to the discerning update strategy which incorporates an information dependent forgetting factor. It is shown that if the input noise is bounded and if the process is stable, then the *a posteriori* prediction error is bounded even without the SPR condition. This is in sharp contrast to the crucial role of the SPR condition in the ELS and output error algorithms.

I. INTRODUCTION

The autoregressive moving average(ARMA) parameter estimation problem arises in many adaptive signal processing applications such as speech processing, seismic data processing and channel equalization. Typically, samples of the measured signal y(t) are modeled as the output of an IIR filter driven by unknown white noise w(t) [1]. The ARMA model is described by the temporal recursion

$$y(t) = a_1\, y(t-1)+..+ a_n\, y(t-n) + w(t) + c_1\, w(t-1) +..+ c_r\, w(t-r) \tag{1}$$

Fitting this ARMA model to the measured data y(t), t =1..T , requires the estimation of the parameters $a_1 .., a_n , c_1 .., c_r$. Recursive schemes like the extended least-squares (ELS), the recursive maximum likelihood (RML) and multi-stage least-squares algorithms have been used to estimate ARMA parameters [2,3]. The ELS algorithm uses the *a posteriori* prediction error $\varepsilon(t)$, as an estimate of w(t). The regressor vector is formed from y(t-1),.., y(t-n) and $\varepsilon(t-1)$,.., $\varepsilon(t-r)$. The standard recursive least-squares (RLS) algorithm is then employed to update the estimates. The algorithm is conceptually simple but restrictive in the sense that convergence of the algorithm can be assured only if the underlying transfer function $H(q^{-1}) = 1/\, C(q^{-1}) - 1/2$ is strictly positive real (SPR), with q^{-1} being the delay operator and

$$C(q^{-1}) = 1 + c_1 q^{-1} + c_2 q^{-2} + .. + c_r q^{-r}$$

The RML algorithm, which uses a filtered version of the regressor vector used in the ELS algorithm, does not require $H(q^{-1})$ to be SPR. However the estimates have to be monitored and projected into a stability region to ensure convergence[2].

In addition to the aforementioned least-squares based methods, there exists a different class of estimation algorithms that estimate membership sets of parameters which are consistent with the measurements, model structure and noise constraints [4]-[8]. These algorithms are particularly useful when the noise distribution is unknown but constraints in the form of bounds on the instantaneous values of the noise are available. The model and noise representations commonly used are

$$y(t) = \theta^{*\,T}\, \Phi(t) + v(t), \quad |v(t)| \le r^{1/2}(t)$$

where {y} is a sequence of scalar observations, θ^* is the parameter vector to be identified, $\Phi(t)$ is a n-vector of variables known at time t, {v} is the noise sequence and $\pm\, r^{1/2}(t)$ are the time varying noise bounds. Given a sequence $\{y(i),\Phi(i)\}$, i=1..k, the optimal membership set

$$\psi^0_{\ k} = \cap^{\ k}_{\ i=1}\, S_i$$

where

$$S_i = \{ \theta : (y(i) - \theta^T \Phi(i))^2 \leq r(i),\ \theta \in \mathbf{R}^n \}$$

From a geometrical point of view S_i is a convex polytope in $\mathbf{R}^n$ and contains the true parameter vector. Finding $\psi^0{}_k$ is computationally intractable and it is therefore necessary to approximate $\psi^0{}_k$ by some set which encloses it tightly and which can be described and updated economically [9]. The different membership-set algorithms differ in the way the optimal membership set is approximated and in the method used to obtain a tight enclosure.

Among these algorithms based on membership sets, a seminal recursive algorithm is the so-called optimal bounding ellipsoid (OBE) algorithm[6-8]. One of the main features of this algorithm is a discerning update strategy. This feature, obtained by the introduction of an information dependent updating/forgetting factor, yields a modular structure thereby increasing the potential for concurrent and pipelined processing of signals. The presence of such a forgetting factor also gives the algorithm the ability to track time varying parameters. The algorithm has the advantageous feature of automatic asymptotic cessation of updates in the fixed parameter case.

In this paper, we extend one of these OBE algorithms[8] to the ARMA case. For the ARMA parameter estimation problem, the OBE algorithm cannot be applied in its present form. However, by assuming that the input white noise is bounded in magnitude, the OBE algorithm can be extended in a manner similar to the ELS algorithm. The convergence analysis of the resulting algorithm, as opposed to that of the ELS algorithm, is deterministic and is performed under the assumptions that the process is stable and that the noise is bounded.The *a posteriori* prediction error is shown to be bounded without imposing any SPR condition. This is in contrast to the convergence analysis of the ELS or output error algorithms in which the SPR condition is used to prove boundedness of the prediction errors and convergence of parameter estimates[10]. By imposing a persistence of excitation condition on the regressor vector, the *a priori* prediction error of the extended OBE algorithm is shown to be bounded and the parameter estimates are shown to converge to a neighborhood of the true parameter vector.

The paper is organized in the following manner. In Section II, a brief review of the OBE algorithm and its properties is presented. In Section III, the algorithm is extended to ARMA parameter estimation. Convergence analysis of the extended algorithm is performed in Section IV. The performance of the algorithm is compared with the ELS algorithm through simulation studies in Section V. Section VI concludes the paper.

II. THE OBE ALGORITHM

Consider the ARX model described by

$$y(t) = a_1\, y(t-1) + .. + a_n\, y(t-n) + b_0 u(t) + b_1\, u(t-1) + .. + b_m\, u(t-m) + v(t)$$

The above equation can be recast as :

$$y(t) = \theta^{*\,T}\, \Phi(t) + v(t) \qquad (2)$$

where

$$\theta^* = [a_1 , a_2 \ \dots\ a_n \ ,\ b_0 , b_1 , \dots, b_m \ ]^T$$

is the vector of true parameters and

$$\Phi(t) = [\ y(t-1), y(t-2), ..\ y(t-n),\ \ u(t), u(t-1), ..\ u(t-m)\ ]^{\,T}$$

is the regressor vector. It is assumed that the noise is uniformly bounded in magnitude, i.e., there exists $\gamma_0 \geq 0$, such that

$$v^{\,2}(t) \ \leq\ \gamma_0{}^2 \qquad \text{for all t, hence}$$

$$(y(t) \ - \ \theta^{*T}\, \Phi(t) \ )^2 \ < \ \gamma_0{}^2$$

Let S_t be a subset of the euclidean space $\mathbf{R}^{\,n+m+1}$, defined by

$$S_t = \{ \theta : (y(t) - \theta^T \ \Phi(t))^2 \leq \gamma_0{}^2 \ , \theta \in \mathbf{R}^{\,n+m+1} \}$$

The OBE algorithm starts off with a large ellipsoid, E_0 , in $\mathbf{R}^{n+m+1}$ which contains all possible values of the modelled parameter θ^*. After the first observation y(1) is acquired, an ellipsoid is found which bounds the intersection of E_0 and the convex polytope S_1.This ellipsoid must be optimal in some sense, say minimum volume[6-7] or by any other criterion[6-8], to hasten convergence. Denoting the optimal ellipsoid by E_1 one can proceed exactly as before with the future observations and obtain a sequence of optimal bounding ellipsoids $\{ E_t \}$. The center of the ellipsoid E_t can be taken as the parameter estimate at the t-th instant and is denoted by $\theta(t)$. If at a particular time instant i, the resulting optimal bounding ellipsoid would be of a " smaller size ", thereby implying that the data point y(i) conveys some " information " regarding the parameter estimates, then the parameters are updated. Otherwise E_i is set equal to E_{i-1} and the parameters are not updated.

Let the ellipsoid E_{t-1} at the (t-1)-th instant be formulated by

$$E_{t-1} = \{ \theta : (\theta - \theta(t-1))^T P^{-1}(t-1) (\theta - \theta(t-1)) \leq \sigma^2(t-1) \}$$

for some positive definite matrix P (t-1) and a non-negative scalar σ^2 (t-1). Then, given y(t), an ellipsoid which bounds $E_{t-1} \cap S_t$ "tightly" is

$$\{ \theta : (1 - \lambda_t) (\theta - \theta(t-1))^T P^{-1}(t-1) (\theta - \theta(t-1)) + \lambda_t (y(t) - \theta^T \Phi(t))^2 \leq (1 - \lambda_t) \sigma^2(t-1) + \lambda_t \gamma_0^2 \} \quad (3)$$

where the forgetting factor $\lambda(t)$ satisfies $0 \leq \lambda(t) < 1$. The size of the bounding ellipsoid is related to the scalar σ^2 (t-1) and the eigenvalues of P(t-1). The update equations for $\theta(t)$, P(t) and $\sigma^2(t)$ are derived in [8]. The optimal ellipsoid which bounds the intersection of E_{t-1} and S_t is defined in terms of an optimal value of λ_t. For the OBE algorithm of [8], the optimum value λ^*_t is determined by minimization of $\sigma^2(t)$ with respect to λ_t at every time instant. The minimization procedure results in λ^*_t being set equal to zero (no update) if

$$\sigma^2(t) + \delta^2(t) \leq \gamma^2(t) \quad (4)$$

If (4) is not satisfied, then the optimal value of λ_t is computed. The parameter estimation procedure is depicted in Fig. 1. An outgrowth of the modular recursive estimation procedure is a parallel-pipelined networking structure [11]. The algorithm is such that the computational complexity of the information evaluation (IE) procedure is much less than that of the updating procedure (UPD). Since, in general, a good number of data samples would be rejected by the IE, both the IE and the UPD would involve significant amounts of idle time. A viable scheme then is to configure a parallel-pipelined network comprising of such modular estimators to process signals from multiple channels. Apart from reducing hardware costs, such a scheme would offer increased reliability since the failure of one UPD processor would not cause any of the channels to fail, in contrast to a system with a dedicated UPD processor for each channel.

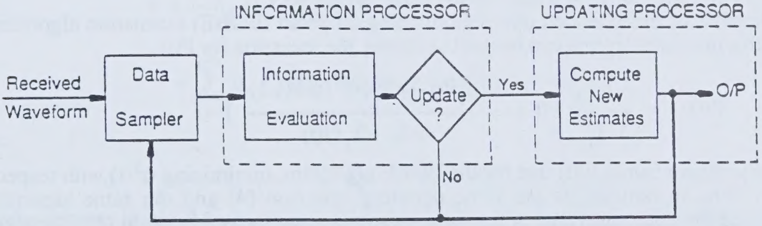


Figure 1 Modular recursive parameter estimator

III. EXTENSION TO ARMA MODELS

The ARMA model described by (1), can be rewritten as

$$w(t) = y(t) - \theta^{*T} \Phi'(t) \qquad (5)$$

where θ^* is the vector of true parameters and is now defined by

$$\theta^* = [a_1 , a_2 \ldots a_n , c_1 , c_2 \ldots, c_r]^T$$

$$\Phi'(t) = [\, y(t-1) , \ldots, y(t-n) , w(t-1) , \ldots, w(t-r) \,]^T$$

Here again, $w(t)$ is assumed to be bounded in magnitude by γ_0. Since the values of the noise sequence $\{w(t)\}$ are not available, the regressor vector $\Phi'(t)$ is not known exactly. If, however, at time t, an estimate of θ^*,

$$\theta(t) = [\, a_1(t), \ldots, a_n(t) \;\; c_1(t), \ldots, c_r(t) \,]^T \qquad (6)$$

is available, $w(t)$ could be estimated by its estimate $\varepsilon(t)$ according to

$$\varepsilon(t) = y(t) - \theta^T(t)\Phi(t)$$

where

$$\Phi(t) = [\, y(t-1) , \ldots y(t-n) , \varepsilon(t-1) , \ldots \varepsilon(t-r) \,]^T \qquad (7)$$

It follows from (5) and the boundedness of $|w(t)|$ that

$$(y(t) - \theta^{*T} \Phi'(t))^2 \le \gamma_0^2$$

Hence if $\varepsilon(t) \approx w(t)$, then $\Phi'(t) \approx \Phi(t)$ and for a suitable $\gamma \ge 0$, such that $\gamma^2 > \gamma_0^2$

$$(y(t) - \theta^{*T} \Phi(t))^2 \le \gamma^2 \qquad (8)$$

Thus $\theta^* \in S_t$, where S_t is defined by

$$S_t = \{\, \theta : (y(t) - \theta^T \Phi(t))^2 \} \le \gamma^2 , \theta \in \mathbf{R}^{n+r} \,\}$$

Hence if the difference between $\varepsilon(t)$ and $w(t)$ is small, applying the OBE algorithm will yield a sequence of bounding ellipsoids $\{E_t\}$ in the parameter space. If (8) holds for all t and $\theta^* \in E_0$, then it is easy to see that $\theta^* \in E_t$ for all time instants t. The optimal bounding ellipsoid E_t is described by

$$E_t = \{\, \theta \in \mathbf{R}^{n+r} : (\theta - \theta(t))^T P^{-1}(t) (\theta - \theta(t)) \le \sigma^2(t) \,\} \qquad (9)$$

and the update equations which follow directly from [8] are

$$P^{-1}(t) = (1 - \lambda_t) P^{-1}(t-1) + \lambda_t \Phi(t) \Phi^T(t) \qquad (10\text{a})$$

$$\theta(t) = \theta(t-1) + \lambda_t P(t)\Phi(t)\, \delta(t) \qquad (10\text{b})$$

$$\delta(t) = y(t) - \theta^T(t-1)\, \Phi(t) \qquad (10\text{c})$$

$$\sigma^2(t) = (1 - \lambda_t)\, \sigma^2(t-1) + \lambda_t \gamma^2 - \frac{\lambda_t (1-\lambda_t)\, \delta^2(t)}{1-\lambda_t + \lambda_t G(t)} \qquad (10\text{d})$$

where

$$G(t) = \Phi^T(t) P(t-1) \Phi(t) \qquad (10\text{e})$$

The above recursive relations along with the initial values

$$P^{-1}(0) = I , \; \theta(0) = 0 \text{ and } \sigma^2(0) = 1/\varepsilon \text{ with } \varepsilon \ll 1$$

form the basis of the extended optimal bounding ellipsoid (EOBE) estimation algorithm[12]. The matrix inversion lemma can be used to obtain the recursion for P(t)

$$P(t) = \frac{1}{1-\lambda_t} \left[\, P(t-1) - \frac{\lambda_t P(t-1)\Phi(t)\Phi^T(t)\, P(t-1)}{1-\lambda_t + \lambda_t G(t)} \,\right] \qquad (11)$$

It is easily shown from (10d) that for the EOBE algorithm, minimizing $\sigma^2(t)$ with respect to λ_t at every time instant yields the same updating criterion (4) and the same algorithm for determining the optimum value of the forgetting / updating factor λ^*_t, as in [8]. The algorithm thus retains the discerning update strategy and the modular adaptive filter structure

Choice of γ^2

Finding a value of γ^2 to ensure that (8) holds for all time instants is not easy. Instead, we will show in the next section that choosing γ^2 to be an upper bound on the square of the magnitude of the output y(t) will ensure that the parameter estimates obtained by applying the EOBE algorithm converge to a neighborhood of the true parameters.

If the ARMA process is stable then if $|w(t)|$ is bounded so is $|y(t)|$. The output process y(t) can be expressed as
$y(t) = h(t) * w(t)$,
where $*$ denotes discrete convolution and h(t) is the equivalent impulse response. Stability of the process implies that there exists a finite M such that

$$\sum_{i=0}^{\infty} | h(i) | \leq M < \infty \tag{12}$$

and if $| w(t) | \leq \gamma_0$ then

$$| y(t) | \leq \sum_{i=0}^{\infty} | h(i) | \, | w(t-i) | \leq M \gamma_o = \gamma \tag{13}$$

In order to obtain a value for γ, the user would therefore need the bound on $|w(t)|$ and an estimate of M. Or alternatively, the outputs could be monitored and a bound on $|y(t)|$ could be obtained before starting the actual parameter estimation procedure. A loose upper bound on either M or the magnitude of the output would suffice since simulations have shown that using values for γ that are several times larger or smaller than the actual bound on $|y(t)|$ has no deleterious effects on the quality of estimates or convergence rate.

IV. CONVERGENCE ANALYSIS

We will first show that if the updating factor sequence is that which minimizes σ^2 at every instant (denoted by $\{ \lambda^*_t \}$) then the parameter estimates converge. To show that, the following lemma will be needed.

Lemma 1. If the magnitude of the noise w(t) is upper bounded and the process is stable then $\sigma^2(t)$, defined in (10d), is always non-negative provided that the initial values $\theta(0)$, $P^{-1}(0)$ and $\sigma^2(0)$ are chosen such that $\theta^T(0)P^{-1}(0)\theta(0) \leq \sigma^2(0)$

Proof. From (10a), (10b) and (10c)

$$\begin{aligned} \theta^T(t) P^{-1}(t) \theta(t) &= (1-\lambda_t) \theta^T(t-1) P^{-1}(t-1) \theta(t-1) + \lambda_t[y^2(t) - \delta^2(t) (1- \lambda_t \Phi^T(t) P(t) \Phi(t))] \\ &= (1- \lambda_1) (1- \lambda_2) \ldots (1-\lambda_t) \theta^T(0) P^{-1}(0) \theta(0) \\ &\quad + \sum_{i=1}^{t} q_{it} [y^2(i) - \delta^2(i) (1 - \lambda_i \Phi^T(i) P(i) \Phi(i))] \end{aligned}$$

where

$$q_{it} = \lambda_i (1- \lambda_{i+1}) \ldots (1- \lambda_t) \geq 0 \quad \forall\, i, t.$$

Since the process is stable and $\{w(t)\}$ is bounded, there exists a γ^2 such that $y^2(t) \leq \gamma^2$
The positive semi-definiteness of $P^{-1}(t)$ will therefore imply that

$$\prod_{i=1}^{t}(1-\lambda_i)\ \theta^T(0)P^{-1}(0)\theta(0) + \sum_{i=1}^{t} q_{it}\,[\gamma^2 - \delta^2(i)\,(1 - \lambda_i \Phi^T(i)\,P(i)\,\Phi(i)\,)] \ \geq 0 \tag{14}$$

But from (11)

$$\lambda_i \Phi^T(i)\,P(i)\,\Phi(i) \ = \ \frac{\lambda_i G(i)}{1-\lambda_i + \lambda_i G(i)} \tag{15}$$

Using (15) in (14) yields

$$\prod_{i=1}^{t}(1-\lambda_i)\ \theta^T(0)P^{-1}(0)\theta(0) + \sum_{i=1}^{t} q_{it}\,[\gamma^2 - \delta^2(i)\frac{1-\lambda_i}{1-\lambda_i + \lambda_i G(i)}\,] \ \geq 0 \tag{16}$$

From (10d), a non-recursive expression for $\sigma^2(t)$ can be obtained as

$$\sigma^2(t) \ = \ (1-\lambda_1)\,(1-\lambda_2)\,...(1-\lambda_t)\,\sigma^2(0) + \sum_{i=1}^{t} q_{it}\,[\,\gamma^2 - \frac{\delta^2(i)\,(\,1-\lambda_i)}{1-\lambda_i + \lambda_i G(i)}\,] \tag{17}$$

Since $\sigma^2(0) \geq \theta^T(0)P^{-1}(0)\theta(0)$, (17) implies that $\sigma^2(t) \geq 0$ for all t . This result will hold for any sequence of forgetting factors $\{\lambda_t\}$ with $0 \leq \lambda_t < 1$.

Theorem 1. If the assumptions of Lemma 1 are satisfied then

1. $$\lim_{t\to\infty} \varepsilon^2(t) \in [\,0, \gamma^2] \tag{18}$$

2. $$\lim_{t\to\infty} \sigma^2(t) \in [\,0, \gamma^2] \tag{19}$$

Proof. See [13].

Note that throughout this section, expressions like (18) should not be taken to mean that $\lim_{t\to\infty}\varepsilon^2(t)$ exists but rather that $\varepsilon^2(t)$ becomes asymptotically less than or equal to γ^2.

The next lemma relates the positive definiteness of $P^{-1}(t)$ to the richness of the regressor vector $\Phi(t)$.

Lemma 2. If there exist positive α_3 and N such that for all t

$$\sum_{i=t}^{t+N} \Phi(i)\,\Phi^T(i) \ \geq \ \alpha_3 I \ > \ 0 \tag{20}$$

then there exists a positive α_4 such that

$$P^{-1}(t) \geq \alpha_4 I > 0$$

Proof of the lemma is the same as that of Theorem 4.1 of [8], it is thus omitted here.

Remark. The positive definiteness of $P^{-1}(t)$ implies that the eigenvalues of P(t) are upper bounded.

Theorem 2. If assumptions of Lemma 1 are satisfied and (20) holds then the EOBE algorithm ensures :

(a) Parameter difference convergence

$$\lim_{t\to\infty} \| \theta(t) - \theta(t-k) \| = 0 \tag{21}$$

for any finite k.

(b) Bounded *a priori* prediction errors

$$\lim_{t\to\infty} \delta^2(t) \in [0, \gamma^2] \tag{22}$$

(c) Bounded parameter misadjustment

$$\lim_{t\to\infty} \| \theta(t) - \theta^* \|^2 \leq M\gamma^2 < \infty \tag{23}$$

for some finite M.

Proof.

(a) From (10b) and (11)

$$\| \theta(t) - \theta(t-1) \|^2 = \frac{\lambda_t^2 \Phi^T(t) P^2(t-1) \Phi(t) \delta^2(t)}{(1 - \lambda_t + \lambda_t G(t))^2}$$

$$\leq e_{max}\{ P(t-1) \} \frac{\lambda_t^2 G(t) \delta^2(t)}{(1 - \lambda_t + \lambda_t G(t))^2} \tag{24}$$

If (20) holds then by Lemma 2, e_{max} { P(t-1) }, the maximum eigenvalue of P(t-1), is bounded for all t. The other term on the right hand side is shown to tend to zero in the proof of Theorem 1. Thus

$$\| \theta(t) - \theta(t-1) \| \to 0 \tag{25}$$

Applying the Minkowski inequality to $\| \theta(t) - \theta(t-k) \|$ and using (25) completes the proof of (21).

(b) From (10e) and (13) it follows that

$$G(t) \leq e_{max} \{P(t-1)\} [\, n\gamma^2 + r \max_{t-r \leq i \leq t-1} \varepsilon^2(i) \,] \tag{26}$$

where n is the order of the AR process and r is the order of the MA process. If (20) holds then e_{max} { P(t-1) }, the maximum eigenvalue of P(t-1), is bounded. Since {ε(t)} is bounded by Theorem 1, {G(t)} is bounded. Then it can be shown (see Theorem 3.2 of [8]), that the *a priori* prediction errors satisfy (22).

(c) Since w(t) and y(t) are bounded by γ and ε(t) is aymptotically bounded by Theorem 1, it can be shown that for large enough t

$$\sum_{l=t}^{t+N} \theta^T(l) \Phi(l) \Phi^T(l) \theta(l) \leq O(\gamma^2) \tag{27}$$

where $\theta(t) = \theta(t) - \theta^*(t)$, and N is given by Lemma 2. Then the result follows by using (21) and (20) in (27) and performing some algebraic manipulations.

Remark: To summarize, the above analysis has shown that if the process is stable and if the driving noise is bounded then the *a posteriori* prediction errors are bounded. In addition if a richness condition is imposed on the regressor vector, then the *a priori* prediction errors are bounded and the parameter estimates are asymptotically contained within a neighborhood of the true parameters.

V. SIMULATION RESULTS

Simulations have been performed to investigate the performance of the EOBE algorithm vis a vis the ELS algorithm. In this paper, we present simulation results for two examples- a broad band ARMA (3,3) process and a non SPR ARMA(3,3) process where the indices p, q in an ARMA(p,q) process refer to the orders of the $A(q^{-1})$ and $C(q^{-1})$ polynomials, respectively.

Example 1. Broad band ARMA (3,3) process
The output data {y(t)} is generated by the following difference equation
$y(t) = - 0.4\, y(t-1) + 0.2\, y(t-2) + 0.6\, y(t-3) + w(t) - 0.6\, w(t-1) + 0.2w(t-2) + 0.4\, w(t-3)$
The noise sequence {w(t)} is generated by a pseudo-random number generator with a uniform probability distribution in [-1.0, 1.0]. The upper bound γ^2 was set equal to 25.0. The parameter estimates were obtained by applying the EOBE algorithm to 1000 point data sequences. One hundred runs of the algorithm were performed on the same model but with different input noise sequences. The average squared parameter error L(t), is computed for the AR coefficients according to the formula

$$L(t) = \frac{1}{100} \sum_{j=1}^{100} l_j(t)$$

where $l_j(t)$, the squared AR parameter error at time t for the j'th run, is defined by

$$l_j(t) = \sum_{i=1}^{n} (a_i(t) - a_i)^2$$

with a_i and $a_i(t)$ being defined by (1) and (6), respectively. The average squared parameter error for the MA coefficients is defined analogously. Fig. 2 displays the average squared estimation errors for AR and MA parameters using the EOBE and ELS algorithms. The figures show that the performance of the two algorithms is comparable. It may be noted that the AR parameter estimates have markedly lower variance (about the true parameters) than the MA parameters. The average number of updates for the EOBE algorithm was 139 for 1000 point data sequences. Thus less than 15% of the samples are used for updates, as compared to the ELS algorithm which updates at every sampling instant.

The tracking capability of the EOBE algorithm(with α=0.2), was compared with that of the ELS algorithm (with forgetting factor=0.99). The same model was used to generate 400 data points. The parameters were then changed by 150% and the next 400 points were generated. Finally the last 200 points were generated by using the original parameters. The average squared parameter error was evaluated over 25 runs and is shown in Figure 3. Even though the formulation of bounding ellipsoids is based on the assumption that the parameters are constant, the simulation results show that the algorithm is able to accommodate changes in model parameters. Analysis of the tracking ability of the algorithm is currently under investigation.

Example 2. Non SPR ARMA(3,3) process

The output data {y(t)} is generated by the following difference equation
$y(t) = - 0.6\, y(t-1) - 0.58\, y(t-2) - 0.464\, y(t-3) + w(t) + 0.2\, w(t-1) + 0.6w(t-2) + 0.2\, w(t-3)$
The noise sequence is generated as in the first example. The upper bound γ^2 was set equal to 3.25. The maximum values (taken over 25 runs of the algorithm) of the squared residual errors, at each iteration were computed and were well within the upper bound γ^2 even though the SPR condition is violated.

VI. CONCLUSION

A recursive parameter estimation algorithm has been extended for ARMA parameter estimation. The main features of the algorithm are a membership set theoretic formulation and a discerning update strategy. Convergence analysis of the algorithm has been performed under the assumptions that the process is stable and that the noise is bounded. The results of the analysis are that the algorithm yields bounded *a posteriori* prediction errors without SPR or persistence of excitation type condition. With a persistence of excitation condition on the regressor vector, boundedness of the *a priori* prediction error can then be established and the parameter estimates are shown to converge to a neighborhood of the true parameters. Simulation results show that the performance of the algorithm is comparable to the ELS algorithm while requiring far fewer updates.

VII. ACKNOWLEDGEMENT

This work has been supported in part by the National Science Foundation under Grant MIP-87-11174; in part by the office of Naval Research under Contract N00014-87-k- 0284, and in part by the National Science Foundation under Grant ECS-8618240.

REFERENCES

[1] B. Friedlander, " System identification techniques for adaptive signal processing, "*IEEE Trans. Acoust. Speech and Signal Process.* , vol. ASSP-30, No. 2, pp. 240-246, April 1982.
[2] L. Ljung and T. Soderstrom,*Theory and Practice of Recursive Identification*, Cambridge, Mass: MIT Press, 1983.
[3] D. Q. Mayne *et al*, " Linear identification of ARMA processes," *Automatica* vol. 18, No. 4, pp. 461-466, July 1982.
[4] F. C. Schweppe, " Recursive state estimation : unknown but bounded errors and system inputs, " *IEEE Trans. Automatic Control* , vol. AC-13, No.1, pp. 22-28, Feb. 1968.
[5] M. Milanese and G. Belaforte, " Estimation theory and uncertainty intervals," *IEEE Trans. Automatic Control* , vol. AC-27, No. 2, pp. 408-414, April 1982.
[6] E. Fogel and Y.F. Huang, " On the value of information in system identification - bounded noise case," *Automatica* , vol. 18, No. 2, pp. 229-238, March 1982.
[7] Y. F. Huang, " A recursive estimation algorithm using selective updating for spectral analysis and adaptive signal processing,"*IEEE Trans. Acoust.Speech and Signal Process.* , vol. ASSP-34, No. 5, pp. 1131-34 , October 1986.
[8] S. Dasgupta and Y.F. Huang, " Asymptotically convergent modified recursive least-squares," *IEEE Trans. Information Theory* , vol. IT-33, No.3, pp. 383-392, May 1987.
[9] J. P. Norton, " Identification and application of bounded-parameter models," *Automatica*, vol. 23, No. 4, pp. 497-507, 1987.
[10] G. C. Goodwin and K.S. Sin, *Adaptive Filtering, Prediction and Control* , Englewood Cliffs, NJ: Prentice-Hall, 1982.
[11] Y. F. Huang and R. W. Liu, " A high level parallel-pipelined network architecture for adaptive signal processing, " in *Proc. 26th IEEE Conf. on Decision and Control*, vol.1, pp. 662-666, Dec. 1987.
[12] A. K. Rao, " Applications and extensions of a novel recursive estimation algorithm with selective updating," M.S. Thesis, Dept. of Electrical & Computer Engineering, University of Notre Dame, Notre Dame, IN. Aug. 1987.
[13] A. K. Rao, Y. F. Huang and S. Dagupta, " ARMA parameter estimation using a novel recursive estimation algorithm with selective updating," submitted for publication to *IEEE Trans. Acoust.Speech and Signal Process.*

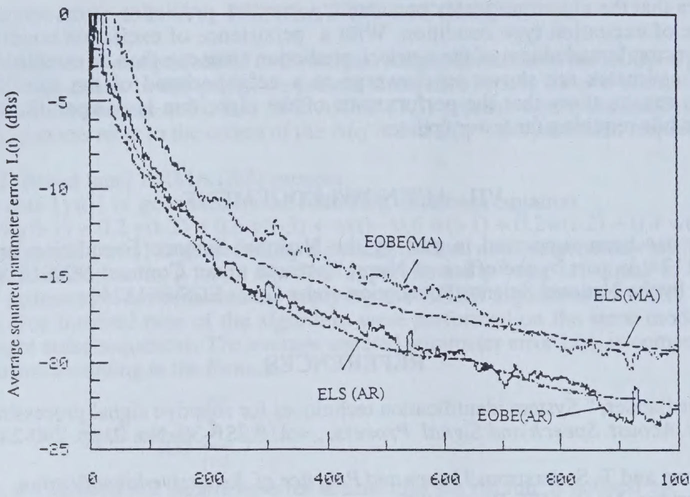


Figure 2. Average squared parameter error for the EOBE and ELS algorithms - Example 1.

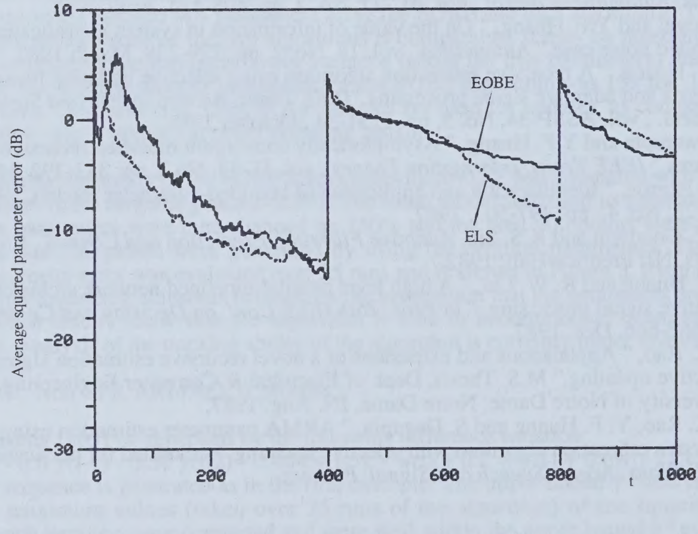


Figure 3. Average squared parameter error for the EOBE and ELS algorithms - Time varying case.

NONLINEAR ITERATIVE TECHNIQUES FOR IMAGE RESTORATION FROM INCOMPLETE SPECTRAL DATA

KIRK D. SANDERS AND ALAN C. BOVIK
Department of Electrical and Computer Engineering
The University of Texas at Austin
Austin, Texas 78712-1084

Abstract - A novel technique for restoring images that have suffered missing cone distortion is introduced. The technique modifies existing spatial-spectral iterative image restoration techniques by the inclusion of nonlinear operations as part of the iterations. Specifically, median filters are applied at each iteration to enhance and accelerate the restoration near edges in the image. Energy equalization within the missing cone is used to further accelerate convergence. Finally, a thresholding strategy is incorporated for restoring binary-valued images that have undergone missing cone distortion. Extensions to gray-level images using threshold decomposition concepts are also discussed.

INTRODUCTION

Many applications require the restoration of multi-dimensional signals with missing spectral data [1]-[5]. Example areas of interest include computer-aided tomography [6], optical serial sectioning light microscopy [7]–[9], and optical sectioning fluorescent microscopy [10], [11] among many others. Typically, the images obtained are missing a large, often biconic region of spatial frequencies (the "missing cone problem"), leaving the spatial image severely distorted. If useful information is to be extracted from such an image, it is necessary to utilize some restoration technique to "fill in" the missing portion of the spectrum.

Numerous techniques have been proposed to solve the missing cone problem [1], [6]. Many use the fact that the Fourier spectrum of any finite object is entire; hence in principle any portion of it can be deduced from knowledge of the spectrum over any finite continuous segment [12]-[14]. Restoration techniques that attempt to exploit this fact include Fourier continuation [12], [14], [15], prolate spheroidal continuation [13], and projection onto convex sets (POCS) [16], [17]. However, most of these approaches present severe difficult difficulties in practical implementation, particularly in the presence of noise: any noise content will contribute to the observed image spectrum, producing distortion in the reconstruction. Thus, more robust restoration methods are desirable.

The well-known Papoulis-Gerchberg iterative, spatial-spectral refinement algorithm ("super-resolution") [4], [5] often proves more practical since it is often effective against noisy data. In this technique the missing cone is not thought of as an area in which the spectral data is missing, but rather as the result of an additive error term with non-zero value only within the missing cone. The algorithm reduces the error term by zeroing the image outside of an assumed boundary, followed by replacement of the known portion of the spectrum. Subsequent iterations progressively reduce the energy of the error term; Tam [18], [19] provides amicable analytic expressions for the process. While the Papoulis-Gerchberg technique is very attractive for its simplicity, it has the drawback of usually requiring many iterations to attain a reasonable restoration.

In this paper, a novel method of missing cone restoration is introduced that modifies the Papoulis-Gerchberg approach. The algorithm is different by the inclusion of nonlinear processing as part of each iteration. In particular, median filtering combined with thresholding and an energy equalization criteria is used for the restoration of images known to be binary-valued. The median filtering step effectively accelerates convergence by stressing the maintenance of image edges in the restoration; thresholding forces the reconstructed image to be binary-valued. The algorithm also differs in that no assumption is made of any object boundary; hence the only *a priori* information (other than that the undistorted image is binary) required is knowledge of the extent of the missing frequencies. Simulations using binary images with missing cone degradation illustrate the feasibility of the approach. The combination of median filtering and thresholding of degraded binary images suggests the use of analogous processing for gray-level images. In particular, the threshold decomposition property of the median filter allows the separation of any quantized gray-level signal into binary components [20]. Thus, it is possible that missing cone restoration may be effected by separately processing each binary component, and recombining the results afterwards as in the threshold decomposition. This possibility is also explored.

MEDIAN FILTERING

The *median filter* [21] belongs to a general class of filters known as order statistic (OS) filters [22], which are non-recursive moving window filters which can be defined on signals of any dimensionality. Thus, suppose that the elements of a finite window centered at signal coordinate i are given by $x_i = \{x_{1:i}, \ldots, x_{N:i}\}$, where N is odd. The *order statistics* of the windowed data values are the algebraically ordered (permuted smallest to largest) values, denoted $x_{(i)} = \{x_{(1):i}, \ldots, x_{(N):i}\}$; hence $x_{(1):i} \le x_{(2):i} \le \ldots \le x_{(N):i}$. The median filter is simply defined by defining the output y_i to be the median value of the windowed data set: $y_i = x_{(M):i}$, where $M = (N+1)/2$. The predominant features of the median filter are that 1) it effectively smooths high-frequency noise, particularly impulse noise, and 2) it does not smooth spatial edges. Thus, incorporating the median filter into the restoration process can be expected to improve convergence near image edges, and to reduce the effects of any high-frequency noise content on the restoration. Conversely, the nonlinearity of the median filter does not guarantee that the algorithm will converge eventually to the undistorted image. However, if the undistorted image does not contain an abundance of detail, then the restoration should be close.

We now consider inclusion of the median filtering operation into the iteration. Modifying Tam's nomenclature [18], [19], define the operators A and B operating on the undistorted image spectrum S(u,v):

$$A[\,S(u,v)\,] = \begin{cases} S(u,v) & ;\ \text{if } (u,v) \text{ is not within the missing cone} \\ 0 & ;\ \text{if } (u,v) \text{ is within the missing cone} \end{cases} \tag{1}$$

and

$$B[\,S(u,v)\,] = \Im\left\{ m[\Im^{-1}S(u,v)] \right\} \tag{2}$$

where $\Im$ and $\Im^{-1}$ represent the Fourier transform and its inverse, and m represents the median filtering operation. The geometries of the missing cone and of the median filter remain unspecified for now. Thus, $A\left[\,S(u,v)\,\right]$ represents the distorted spectral image with a missing cone of data. A possible refinement of the image may be obtained by the operation:

$$S^{(1)}(u,v) = A[\,S(u,v)\,] + (I - A)\Big[\,B[\,A[\,S(u,v)\,]\,]\,\Big] \tag{3}$$

where I is the identity operation on S(u, v). Iterating yields

$$S^{(n)}(u,v) = A[\,S(u,v)\,] + (I - A)\,[\,B[\,S^{(n-1)}(u,v)\,]\,] \tag{4}$$

for n ≥ 2. The operator B is defined using an unspecified median filter; generally, successive iterations may use a different shape and/or size of the median filter window. While the iteration (4) is of interest, we have found that restoration can be further accelerated by altering the result of the first stage of the algorithm.

ENERGY EQUALIZATION WITHIN THE MISSING CONE

Consider the initial image refinement (3); it is desirable that this estimate (and all subsequent iterations) should be as close as possible to the original, undistorted image. Thus, within the boundaries of the original missing cone, the newly-introduced spectral data should resemble the original spectra as closely as possible. It is assumed here that the missing spectra did not contain a disproportionate amount of the total image power. This assumption is often quite reasonable since it is satisfied by the images of many real-world objects. Using this assumption, the average energy within the missing region of frequencies can be expected to be of the same order of magnitude as the average energy outside of the missing cone. By enforcing this constraint artificially, the rate of convergence can be accelerated; otherwise, the average energy within the missing cone region is generally many magnitudes smaller than that outside the missing cone during the early iterations of the algorithm.

The following technique has proved effective for equalizing the image energy within the missing cone following the first iteration (3). The technique involves sectioning $S^{(1)}(u,v)$ into concentric rings of radial width r, and calculating the energy contained in each ring separately. The innermost region over which the energies are averaged is a circle of radius r. Let the concentric rings be numbered by their distance from the center, defining ring $\mathfrak{R}_0$ as the central circle of radius r, and $\mathfrak{R}_n$ denoting the nth ring outward. Then spectral coordinate (u, v) is contained in $\mathfrak{R}_q$, where $q = \mathrm{INT}\left[\,\sqrt{(u/r)^2+(v/r)^2}\,\right]$. Now let

$$P_{q1} = \mathrm{AVE}\left[\,S^{(1)}(u, v)\,\right]^2 \tag{5}$$

where the average is calculated over all values $S^{(1)}(u, v)$ such that (u, v) lies in the intersection of $\mathfrak{R}_q$ with the missing cone. Similarly,

$$P_{q2} = \mathrm{AVE}\left[\,S^{(1)}(u, v)\,\right]^2 \tag{6}$$

where the average is calculated over all values $S^{(1)}(u, v)$ such that (u, v) lies in the intersection of $\mathfrak{R}_q$ with the region of known frequencies. These quantities are then used to define the following operation:

$$\varphi\left[\,S^{(1)}(u, v)\,\right] = S^{(1)}(u, v) \cdot \sqrt{\frac{P_{q2}}{P_{q1}}} \tag{7}$$

where (u, v) lies in the intersection of $\mathfrak{R}_q$ with the missing cone. Whenever (u, v) lies outside of the missing cone, φ is defined as the identity operator.

The revised first estimate of the restored image is subsequently $\varphi\left[\,S^{(1)}(u, v)\,\right]$. By amplifying the newly-added spectral data within the missing cone to a level similar to the known frequencies, a better early estimate is generally obtained, thus accelerating the iteration process. We have found that it suffices to apply the operator φ only once, since subsequent iterations do not reduce meaningful energy within the missing cone.

Care must be taken in choosing the radial width of the rings over which the frequencies are equalized. If the ring width is too small, the averages obtained are taken over an insufficient number of frequency samples, which leads to unpredictable low-frequency artifacts in the restoration. If the ring width is too large, lower frequencies (over which much of the image energy resides) will dominate: this tends to exaggerate the higher frequencies leading to high-frequency noise artifacts in the restoration. A ring width equal to approximately 1/64 of the image extent (512 x 512 in our experiments) has proved effective.

MEDIAN FILTER GEOMETRIES

The shape and size of the median filter window used in each iteration has been found to effect the restoration process considerably. Clearly, the filter geometry should contribute towards the restoration process as much as possible. Unfortunately, very little theoretical work on 2-D median filters exists. Hence, the choice of the filter window geometry proceeds along somewhat heuristic lines.

Good results were obtained by alternating between two different rectangular filter windows elongated in the direction of the missing cone. By example, suppose that F(u,v) is a spectral image that has suffered a missing cone centered about ray C, which passes through (0,0) at an angle α with the u–axis. Similarly, define ray C' as the ray in the x–y (image) plane passing through the origin at an angle α with the x–axis. The median filter window $W_{i,j}$ is then defined as being rectangular in shape and having a width of i pixels perpendicular to ray C', and a length of j pixels parallel to ray C'. In the restoration of the images of size 512 x 512, good results were obtained by alternating the use of filter windows $W_{5,21}$ and $W_{9,17}$; i.e., filtering with $W_{5,21}$ in the first iteration, filtering with $W_{9,17}$ in the second iteration, etc.

Thin, rectangular windows oriented parallel to ray C' were chosen since square filter windows were found to introduce new distortions into the spatial image: high-curvature object boundaries were excessively blurred. However, non-square 2-D median filters generally leave unavoidable "streaks," particularly if applied repeatedly [23]. It was found that by alternating the rectangular filters having different proportions, the streaking problem is avoided while high-curvature contours are still retained.

THRESHOLDING

If the original, undistorted image is known to be binary-valued or nearly binary-valued, additional processing can be used to further accelerate the convergence of the algorithm and ensure that the restored image is also nearly binary. While it is intended to extend the concepts outlined here for the restoration of general gray-level images, there are applications requiring the restoration of missing spectral data from images that are nearly binary or of very high contrast. One application where a binary-valued restoration is often satisfactory is optical serial-sectioning fluorescent microscopy, where objects of interest are highlighted artificially against a dark background.

Here the operator B defined by (2) is modified to constrain the restored image at the nth iteration $\mathfrak{F}^{-1}S^{(n)}(u,v)$ to be binary. Note that if $B\,[\,S^{(n-1)}(u,v)\,] = S(u,v)$, then $S^{(n)}(u,v) = S(u,v)$, i.e., perfect restoration has been achieved. Assuming that the undistorted image S(u,v), is binary, it is desired that $\mathfrak{F}^{-1}\big[B\,[\,S^{(n-1)}(u,v)\,]\big] = m\big[\mathfrak{F}^{-1}[\,S^{(n-1)}(u,v)\,]\big]$ be binary as well. In the following a thresholding operator τ_n is defined on $m\big[\mathfrak{F}^{-1}[\,S^{(n-1)}(u,v)\,]\big]$ to constrain the restoration towards a binary result.

Missing cone distortion will generally transform a binary image into a grey-level image. Moreover, the histogram of the distorted image may not be bimodal. Therefore, in the first stage of restoration define the operator τ_1 on the spatial image $f(x,y) = m\big[\mathfrak{F}^{-1}[\,A\,S(u,v)\,]\big]$ as follows:

$$\tau_1[\,f(x,y)\,] = \begin{cases} Z_2 \;; & \text{if } f(x,y) \geq M \\ Z_1 \;; & \text{if } f(x,y) < M \end{cases} \tag{8}$$

where Z_1 and Z_2 are the minimum and maximum grey-levels, respectively, that are possible in f(x, y) and $M = (Z_1 + Z_2)/2$. The first iteration then becomes:

$$S^{(1)}(u,v) = A\,[\,S(u,v)\,] + (I - A)\,\big[\,B_1[\,A\,[\,S(u,v)\,]\,]\,\big] \tag{9}$$

where B_1 is defined by:

$$B_1[\,A[\,S(u,v)\,]\,] = \Im\Big\{\tau_1\{\,m[\,\Im^{-1}A[\,S(u,v)\,]\,]\,\}\Big\} \tag{10}$$

Incorporating the operator τ_1 into the first iteration produces the result $F_1(u, v) = \Im^{-1}[\,S^{(1)}(u, v)\,]$ which is roughly bimodal.

Although the τ_1 operator effectively forces bimodality on the result of the first iteration, severe distortions result if it is used in subsequent iterations. Better results are obtained in subsequent iterations by emphasizing bimodality in $F_1(u, v)$. Hence define G_m as the spatial image that results by zeroing every pixel in $F_1(u, v)$ having grey-level value occurring less than m times in $F_1(u, v)$. Further define m' as the smallest m such that $G_{m'}$ is strictly bimodal, and let L be the largest grey-level in $F_1(u, v)$ such that L occurs more than m' times in $F_1(u, v)$ and $L < (Z_1 + Z_2)/2$. Similarly, define U as the smallest grey-level found in $F_1(u, v)$ such that U occurs more than m' times in $F_1(u, v)$ and $U \geq (Z_1 + Z_2)/2$.

The operator τ_n operating on the spatial image $f_n(x, y) = \Im^{-1}[\,S^{(n-1)}(u, v)\,]$ (for $n \geq 2$) is defined as:

$$\tau_n[\,f(x,y)\,] = \begin{cases} Z_2\ ; & \text{if } f(x,y) \geq Z_U \\ f(x,y)\ ; & \text{if } Z_L < f(x,y) < Z_U \\ Z_1\ ; & \text{if } f(x,y) \leq Z_L \end{cases} \tag{11}$$

where

$$Z_U = U - (n-2)k \tag{12}$$

and

$$Z_L = L + (n-2)k \tag{13}$$

for some specified integer k. The nth iteration is then:

$$S^{(n)}(u,v) = A[\,S(u,v)\,] + (I - A)\Big[\,B_n[\,S^{(n-1)}(u,v)\,]\,\Big] \tag{14}$$

where the operator B_n is defined as:

$$B_n[\,S^{(n-1)}(u,v)\,] = \Im\Big\{\tau_n\{m[\Im^{-1}[\,S^{(n-1)}(u,v)\,]\,]\}\Big\}. \tag{15}$$

The parameter k determines the speed at which the algorithm constrains the restoration estimates toward a binary image. In fact, k determines the number of required iterations. Indeed, the final restored image is $S^{(K)}(u, v)$, where

$$K = 2 + \mathrm{INT}\Big[\,(U - L)/2k\,\Big]. \tag{16}$$

Thus, the parameter k is inversely proportional to the number of required iterations.

EXAMPLES

The iterative restoration algorithm was applied to two binary images suffering missing cone distortion. Fig. 1a is an image of several concentric rings surrounding squares of differing orientations; Fig. 2a contains a number of geometric figures. Each of the images was designed to contain significant edges along a continuum of orientations. Figs. 1b and 2b show the same images after undergoing missing cone distortion. The percentage of frequencies removed from the images was 22% and 33%, respectively; the gray level range extends from 0 to 256. In each case a biconic region of frequencies (bounded by rays emanating from the frequency origin) spanning vertical orientations was zeroed. The degree of detail missing in the distorted images is significant, due to the total loss of edge gradient information along the directions of the missing frequencies. Figures 1c and 2c show the result of the restoration algorithm using iteration numbers k = 7 and k = 6, respectively; the corresponding number of iterations was 5 and 7. Considering the small number of iterations required, the restorations are impressive.

DISCUSSION

Given the success of the restoration technique as applied to binary images, it is natural to consider whether the it can be extended for the restoration of gray-level images suffering missing distortion. One possibility is to apply the same algorithm to a set of images obtained via a "threshold decomposition" of the gray-level image.

Fitch *et al.* [20] have shown that any quantized (to N levels) discrete signal f can be decomposed into N binary signals $\{f_1, f_2, \ldots, f_N\}$ such that $f = \sum_{i=1}^{N} f_i$, and such that $m[\,f\,] = \sum_{i=1}^{N} m[\,f_i\,]$, where m is any median filtering operation. The signals f_i are determined by the thresholding operation:

$$f_i(x) = \begin{cases} 1 \; ; & \text{if } f(x) \geq i \\ 0 \; ; & \text{if } f(x) < i \end{cases} \qquad (17)$$

for each gray-level i.

The question arises whether the restoration technique may be applied to each of the resultant binary images, producing a set of improved binary images that can be summed to yield an effective overall restoration. The possibility is further motivated by the fact that the spectra of the decomposed binary images sum to the spectrum of the original gray-level image.

While not detailed here, we have had limited success with direct application of the technique to gray-level images. Briefly, while there is a definite improvement in most images, we have found that occasional artifacts develop which appear to emanate from objects. The resolution to this problem probably requires modification of the thresholding operators (8) and (11). That this is the case may be deduced, since the decomposed binary images defined by (17) display histograms of decreasing densities of 1's. Thus, the thresholding process should be defined to accommodate the variable histogram characteristics as a function of the level in the threshold decomposition. An obvious drawback of the threshold decomposition approach is the computational requirement: the number of computations increases by a factor equal to the number of quantization levels. Clearly, special-purpose hardware would be required to obtain results within a reasonable time; given that, the small number of iterations required will expedite processing. However, the problem may be simplified somewhat by reducing the degree of the decomposition at the expense of some gray-level resolution. In particular, the decomposition (17) may be defined using any subset of the gray-level set. Further experiments along these lines will be developed in later reports.

REFERENCES

[1] T.S. Huang, Ed., *Image Reconstruction from Incomplete Observations,* JAI Press, 1984.

[2] A. Rosenfeld and A.C. Kak, *Digital Picture Processing,* New York, Academic Press, 1982.

[3] D.C. Munson, Jr. and J.L.C. Sanz, "Phase-only Image Reconstruction from Offset Fourier Data," *Opt. Eng.*, vol. 25, 1986.

[4] A. Papoulis, "A New Method of Image Restoration," Joint Services Technical Advisory Committee Rept. no. 39, 1973.

[5] R.W. Gerchberg, "Super-Resolution through Error Energy Reduction," *Opt. Acta,* vol. 21, 1974.

[6] R. Rangayyan, A.P. Dhawan, and R. Gordon, "Algorithms for Limited-View Computed Tomography: An Annotated Bibliography and a Challenge," *Appl. Opt.*, vol. 24, 1985.

[7] N. Streibl, "Three-Dimensional Imaging by a Microscope," *J. Opt. Soc. Am. A.*, vol. 2, 1985.

[8] A. Erhardt, G. Zinser, D. Komitowski, and J. Bille, "Reconstructing 3-D Light- Microscopic Images by Digital Image Processing," *Appl. Opt.*, vol. 24, 1985.

[9] F. Macías-Garza, A.C. Bovik, K.R. Diller, S.J. Aggarwal, and J.K. Aggarwal, "Digital Reconstruction of Three-Dimensional Serially Sectioned Optical Images," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-36, July 1988.

[10] F.S. Fay and K.E. Fogarty, "Analysis of Molecular Distribution in Single Cells using a Digital Imaging Microscope," in *Optical Methods in Cell Physiology*, P.E. Weer and B. Salzberg, Ed., Wiley, New York, 1986.

[11] K.E. Fogarty and W. Carrington, "3-D Molecular Distribution in Living Cells by Deconvolution of Optical Serial Sections using Light Microscopy," in *Proc. 13th Ann. Northwest Bioeng. Conf.*, K.R. Foster, Ed., 1987.

[12] T. Inouye, "Image Reconstruction with Limited Angle Projection Data," *IEEE Trans. Nucl. Sci.*, vol. NS-26, 1979.

[13] B. R. Frieden, "Band-Unlimited Reconstruction of Optical Objects and Spectra," *J. Opt. Soc. Am.*, vol. 57, 1967.

[14] J.L. Harris, "Diffraction and Resolving Power," *J. Opt. Soc. Am.*, vol. 54, 1964.

[15] J.L.C. Sanz and T.S. Huang, "Support-Limited Signal and Image Extrapolation," in *Image Reconstruction from Incomplete Observations*, T.S. Huang, Ed., JAI Press, pp. 1–82, 1984.

[16] D.C. Youla and J. Webb, "Image Restoration by the Method of Convex Projections: Part 1--Theory," *IEEE Trans. Med. Imaging*, vol. MI-1, 1982.

[17] M.I. Sezan and H. Stark, "Image Restoration by the Method of Convex Projections: Part 2--Applications and Numerical Results," *IEEE Trans. Med. Imaging*, vol. MI-1, 1982.

[18] K.C. Tam, V. Perez-Mendez, and B. MacDonald, "Limited Angle 3-D Reconstructions from Continuous and Pinhole Projections," *IEEE Trans. Nucl. Sci..*, vol. NS-27, 1980.

[19] K.C. Tam, "Limits to Image Reconstruction from Restricted Angular Input," *IEEE Trans. Nucl. Sci..*, vol. NS-28, 1981.

[20] J.P. Fitch, E.J. Coyle, and N.C. Gallagher, Jr., "Median Filtering by Threshold Decomposition," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-32, 1984.

[21] N.C. Gallagher, Jr. and G.L. Wise, "A Theoretical Analysis of the Properties of Median Filters," *IEEE Trans. Acoust. Speech Signal Proces.*, vol. ASSP-29, 1981.

[22] A.C. Bovik, T.S. Huang, and D.C. Munson, Jr., "A Generalization of Median Filtering Using Linear Combinations of Order Statistics," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-31, Dec. 1983.

[23] A.C. Bovik, "Streaking in Median Filtered Images," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-35, Mar. 1987.

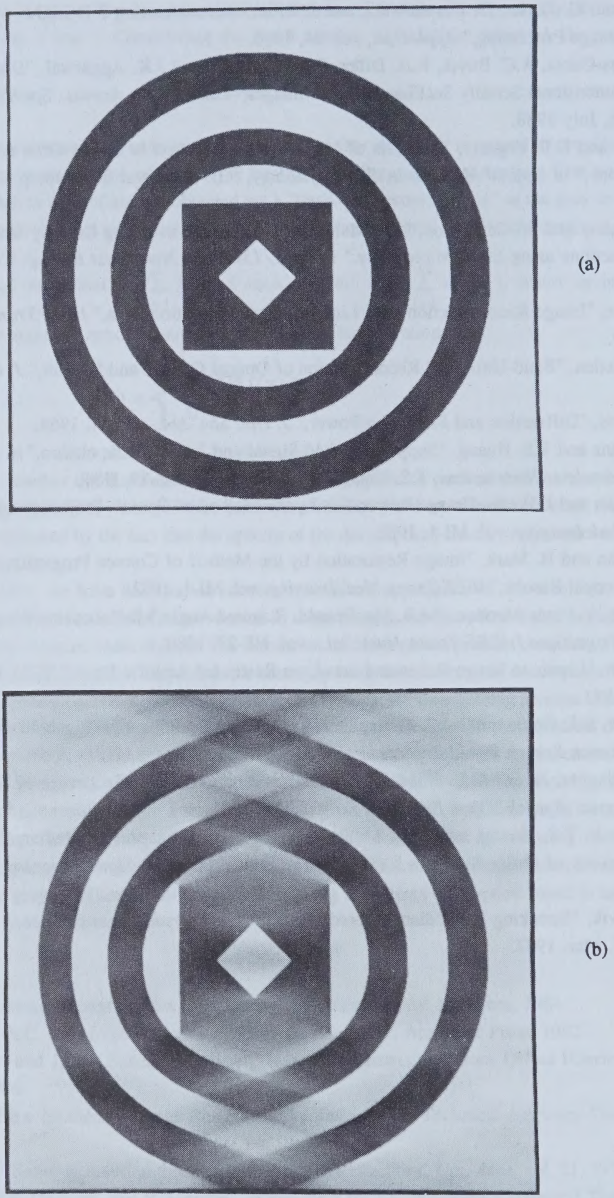
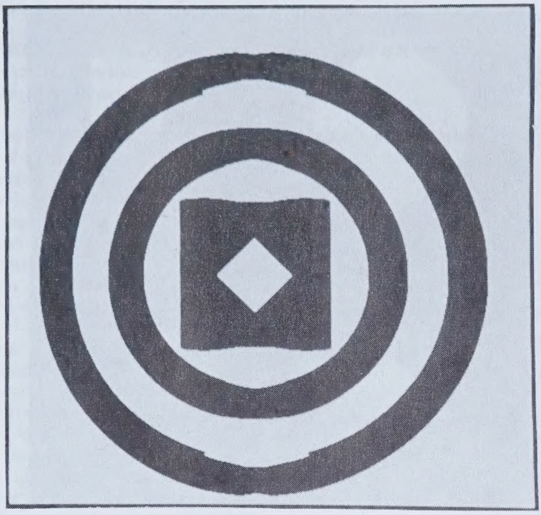
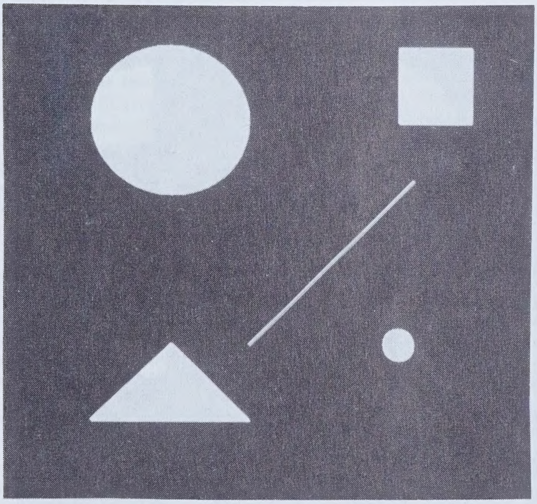


Fig. 1. (a) Binary image of concentric figures; (b) the same image after undergoing 22% missing cone distortion; (c) reconstructed image.



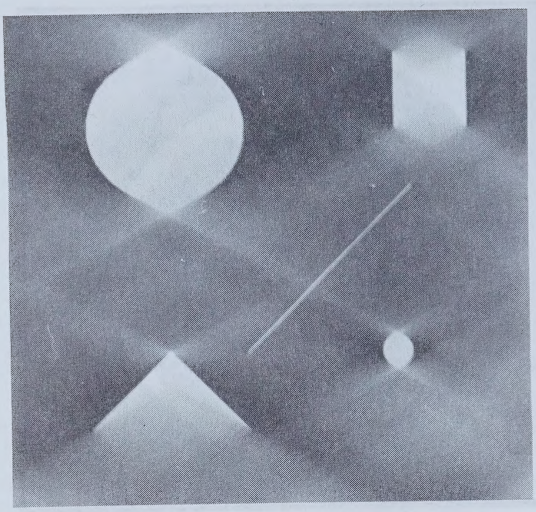
(c)

Fig. 1. (Cont'd).

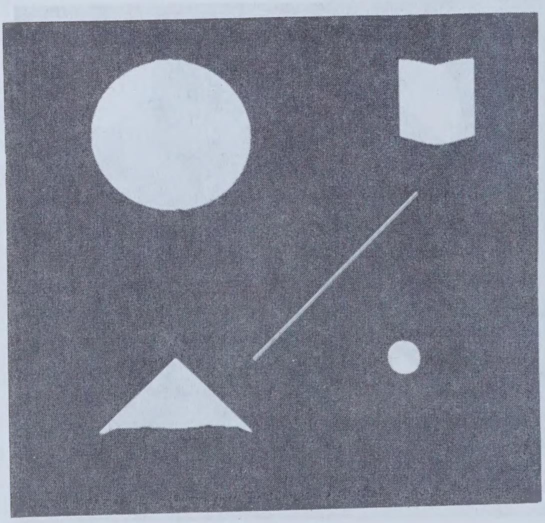


(a)

Fig. 2. (a) Binary image of various geometric figures; (b) the same image after undergoing 33% missing cone distortion; (c) reconstructed image.



(b)



(c)

Fig. 2. (Cont'd).

EIGENANALYSIS BASED BISPECTRUM ESTIMATION

DAVID L. SHERMAN and MICHAEL D. ZOLTOWSKI
School of Electrical Engineering
Purdue University, West Lafayette, IN 47907

Abstract: A new technique for the estimation of the bispectrum of phase coupled sinusoids in white noise is introduced. This eigenstructure estimator divides triple correlation information into two vector subspaces, the signal and noise subspaces for robust determination of coupled frequency components. This method uses full 2-D statistics from a single, non-redundant hextant of the third order cumulant domain.

The invention of high resolution techniques for the estimation of the power spectral density have provided impetus for work in developing procedures for their implementation in finding the bispectrum of a stochastic process. Already, parametric methods such as the autoregressive method of Raghuveer and Nikias are receiving intensive study [1]. Then An, et. al. [4] proposed an optimized, parametric bispectrum estimator which uses an overdetermined, system of equations over the full triple correlation region of support to estimate AR parameters.

For conventional power spectrum estimation even higher resolution performance and better frequency estimation is accomplished with one of the eigendecomposition techniques of power spectral estimation. MUSIC (muliple signal classification) is a very popular spectral analysis technique that requires the estimation of the autocorrelation matrix of a random process assumed to be composed of a known number of complex sinusoidal signals immersed in white noise [3]. The key to improved performance of these methods is the division of information in the autocorrelation matrix into two vector subspaces, the signal subspace and the orthogonal (or noise) subspace. The dimensionality of the signal subspace is determined by the number of assumed complex sinusoids immersed in the noise sequence in question. Eigenanalysis allows for removal of extraneous noise contributions for a much higher signal to noise ratio in the ultimate calculation of the frequencies of the embedded sinsuoids. The success in subspace approaches for power spectral density estimation have motivated research in applying them to the bispectrum and the investigation of non-linear interactions among harmonic components of a time series. Giannakis [2], Swami and Mendel [5] have utilized rank reduction techniques on higher order statistics for robust system identification. In this study we develop a MUSIC type algorithm for the estimation of the bispectrum of phase coupled sinusoids.

A random process consisting of M unit amplitude, real sinusoids of radian frequency, μ_i , i=1,2...M, of random phase in white noise has a standard autocorrelation function,

$$r_{xx}(k) = \frac{1}{2}\sum_{i=1}^{M}\cos(\mu_i kT)+\rho_i\delta(k) \qquad (1)$$

where ρ_i is the noise variance and $\delta(k)$ is the Kronecker delta. The (px1) x (p+1) autocorrelation matrix, where $p > 2M$, can be subsequently be expressed as the sum of corresponding signal and noise autocorrelation matrices:

$$R_p = S_p+W_p$$

Here S_p and W_p are the signal and noise autocorrelation matrices, respectively. The signal matrix will have the eigendecomposition

$$S_p = \sum_{i=1}^{p+1}\lambda_i v_i v_i^H$$

where λ_i denotes the ith eigenvalue and v_i, the respective eigenvector and $\lambda_1 \leq \lambda_2 \leq \ldots \lambda_{2M}$. It can be shown that the signal matrix will have exactly 2M non-zero eigenvalues. The corresponding eigenvectors known as the principal eigenvectors will span the same subspace as the M real sinusoids comprising the signal portion of the time series. The remaining p + 1 - 2M eigenvectors of the signal matrix span the noise subspace. The spectral estimator makes use of the orthogonality between both subspaces. The noise subspace eigenvectors will be orthogonal to any vectors spanning the signal subspace, e.g. sine waves comprising the signal.

To develop eigendecomposition for bispectral analysis, we first define bispectrum as the Fourier transform of the third order cumulant or autocorrelation sequence,

$$B(\omega_1,\omega_2) = \sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} c(m,n)\exp[-j(\omega_1 m+\omega_2 n)]$$

where

$$c(m,n) = E\{x(k)x(k+m)x(k+n)\}.$$

The third order autocorrelation obeys the following symmetry properties [1]:

$$c(m,n) = c(n-m,-m) = c(-m,n-m)$$
$$= c(m-n,-n) = c(-n,m-n) = c(n,m) \qquad (2)$$

The bispectrum can detect quadratic nonlinearity among a triple of sinusids. Three sinusoids are said to be quadratically phase coupled if one component's frequency and phase are the sum or difference of the two other components' frequency and phase. In the bispectral domain a single peak occurs at coordinates (ν_1, ν_2) only if sinusoids with frequencies ν_1, ν_2 and $(\nu_1 + \nu_2)$ exist and all these components are phase locked.

Similar to the case of power spectral density estimation, the third order autocorrelation function for N sets of phase coupled unit amplitude sinusoid triples in non-zero third moment white noise is,

$$\begin{aligned} c(m,n) = \frac{1}{4}\sum_{i=1}^{N} &\cos(\nu_{i2}m+\nu_{i1}n)+\cos(\nu_{i3}m+\nu_{i1}n) \\ &+\cos(\nu_{i1}m+\nu_{i2}n)+\cos(\nu_{i3}m-\nu_{i2}n) \\ &+\cos(\nu_{i1}m-\nu_{i3}n)+\cos(\nu_{i2}m-\nu_{i3}n) \\ &+\beta\delta(m,n) \end{aligned} \tag{3}$$

where ν_{ij} is the jth individual sinusoid of the ith set. β is the magnitude of the third moment of the random process.

We may then arrange the third order cumulant sequence in a $(p+1)^2 x (p+1)^2$ order block Toeplitz matrix, $\underline{\mathbf{C}}$, arranged thusly:

$$\underline{\mathbf{C}} = \begin{bmatrix} C(0) & C(1) & \cdots & C(p) \\ C(-1) & C(0) & \cdots & C(p-1) \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ C(-p) & C(-p+1) & \cdots & C(0) \end{bmatrix} \tag{4}$$

Each constituent block entry, C(i), is a (p+1)x(p+1) matrix which is also Toeplitz and is filled with third order cumulant entries,

$$C(i) = \begin{bmatrix} c(i,0) & c(i,1) & \cdots & c(i,p) \\ c(i,-1) & c(i,0) & \cdots & c(i,p-1) \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ c(i,-p) & c(i,-p+1) & \cdots & c(i,0) \end{bmatrix}$$

This doubly Toeplitz matrix is asymptotically hermitian for two reasons. First of all, we note that c(m,n) = c(-m,-n) if we apply $\cos(\theta) = \cos(-\theta)$ to (3) above. Secondly, we can apply the symmetry properties in (2) to $\underline{\mathbf{C}}$. For each of the sinusoid terms in the expected cumulant sequence we expect a pair of non-zero eigenvalues, hence the rank of the matrix is 12N. We need only calculate third order cumulants in the non-redundant hextant of the third order cumulant domain, i,e. $0 \leq m,n \leq p$ and $m \leq n$. The corresponding spectral estimator is,

$$B(\omega_1,\omega_2) = \left(\underline{\mathbf{e}}^H(\omega_1,\omega_2) \sum_{k=12N+1}^{p} \mathbf{v}_k\right)^{-1} \tag{5}$$

where the eigenvectors are again ordered by respective decreasing eigenvalue, i.e. the eigenvectors comprising the projection operator span the noise subspace. The kernel or complex sinusoid block vector, $\underline{\mathbf{e}}(\omega_1,\omega_2)$, is defined as,

$$\underline{\mathbf{e}}(\omega_1,\omega_1) = \begin{bmatrix} \mathbf{e}_0(\omega_1,\omega_2) \\ \mathbf{e}_1(\omega_1,\omega_2) \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{e}_p(\omega_1,\omega_2) \end{bmatrix} \tag{6}$$

Each block vector, $\mathbf{e}_k(\omega_1,\omega_2)$, is subsequently,

$$\mathbf{e}_k(\omega_1,\omega_2) = \exp(j\omega_1 kT) \begin{bmatrix} 1 \\ \exp(j\omega_2 T) \\ \cdot \\ \cdot \\ \exp(j\omega_2 pT) \end{bmatrix}$$

for k = 1, 2,...p. Due to the orthogonality of signal and noise subspace, the projection of the frequency kernel onto the noise subspace will be small whenever the this kernel assumes the values of constitutent phase locked frequencies. Therefore a peak will appear in the bispectrum.

We have run simulations of this MUSIC-based estimator using noiseless, Gaussian noise and high third moment non-Gaussian noise. Studies have shown that the estimator correctly generates bispectral magnitudes for all three noise cases for up to three sets of phase coupled triples.

References

1. M. R. Raghuveer and C. L. Nikias, "Bispectrum Estimation: A Parametric Approach," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-33, pp. 1213-1230, Oct. 1985

2. G. B. Giannakis, "Signal Processing Via Higher Higher-Order Statistics," Ph.D. dissertation, University of Southern California, Los Angeles, July 1986

3. R. O. Schmidt, "Multiple Emitter Location and Signal Parameter Estimation," *IEEE Trans. Antennas Propagat.,* vol. AP-34, pp. 276-280, March 1986

4. C. K. An, S. B. Kim, and E. J. Powers, "Optimized Parametric Bispectrum," *Proc. IEEE Int. Conf. ASSP,* New York, NY, p. 2392-2395, April 1988

5. A. Swami and J.M. Mendel, "Cumulant-Based Apporach to the Harmonic Retrieval Problem," *Proc. IEEE Int. Conf. ASSP,* p. 2264-2267, New York, NY, April 1988

OPTIMAL DISCRIMINATION OF MULTIDIMENSIONAL AUTOREGRESSIVE PROCESSES

Milan M. Milosavljevic
Institute of Applied Mathematics and Electronics, Belgrade, Yugoslavia
School of Electrical Engineering, Belgrade University, Yugoslavia
School of Electrical Engineering, Cornell University, Ithaca, NY 14853

I. PROBLEM STATEMEN AND MAIN RESULTST

This paper considers a class of p-dimensional discrete processes $\{X_n\}$, generated by the multidimensional autoregressive model of order r :

$$X_n = \sum_{i=1}^{r} A_i X_{n-i} + E_n \tag{1}$$

where A_i, i=1,...,r are the p x p matrices and $\{E_n\}$ is a sequence of mutually independent normally distributed p-dimensional random vectors with mathematical expectation $\mathbf{E}\{E_n\}=0$ and covariance matrix $\mathbf{E}\{E_n E_n^T\}=D$. It is assumed that process $\{X_n\}$ is strictly stationary, i.e., the roots of the characteristic equation

$$\left| I_p - \sum_{i=1}^{r} A_i z^{-i} \right| = 0 \tag{2}$$

lie inside the unit circle, where I_p denotes unit p x p matrix.

The classification system for processes generated by the model (1) is defined as follows.

A1. On the set of processes $\{X_n\}$, Q mutually exclusive classes ω_i, i=1,...,Q, are defined and we say that a process $\{X_n\}$ belongs to the class ω_i, if and only if, it is generated by the model (1) with parameters $\Theta_i=\{A_{1i}, A_{2i}, ..., A_{r_i i}, D_i\}$, where r_i denotes the order of the model (1) for class ω_i.

A2. A priori probabilities of occurance of sequences from class ω_i, i=1,2,...,Q are equal to $P(\omega_i)$, i=1,2,...,Q, $\sum_{i=1}^{Q} P(\omega_i) = 1$.

A3. The classification system works under complete a priori uncertainty, i.e., parameters Θ_i, i=1,2,...,Q, that define classes, are unknown. But at system's disposal there are the training sets Y_i, of length n(i), for each class ω_i :

$$Y_i = \left[Y_1^{(i)} \;\; Y_2^{(i)} \;\; \cdots \;\; Y_{n(i)}^{(i)} \right] , \quad Y_i \in \omega_i , \; i=1,2,...,Q. \tag{3}$$

A4. The first r_i observations $X_1, X_2, \dots, X_{r_i}$, considered as the initial conditions for the model (1), do not give any information about the parameters Θ_i. This condition we can express as :

$$f(\Theta_i \mid X_1, X_2, ..., X_{r_i}) = f(\Theta_i) , \; i=1,2,...,Q. \tag{4}$$

Synthesis of the classification system that minimizes the average probability of error, when conditioned on given training samples, leads to the decision rule :

$$X \in \omega_{i^*} = i^* = arg\left\{ \max f(X \mid \omega_i, Y_i) \, P(\omega_i) \right\}, \tag{5}$$

where the class conditional probability density functions are given by integration

$$f(X \mid \omega_i, Y_i) = \int f(x \mid \omega_i, \Theta_i) \, f(\Theta_i \mid Y_i) \, d\Theta_i \; . \tag{6}$$

A5. To find a posteriori probability density function of Θ_i given the training samples Y_i, we have to know its a prior (initial) probability density function $f(\Theta_i)$. According to the invariance theory due to Jeffreys , we adopt for the $f(\Theta_i)$:

$$f(A_{1i}, A_{2i}, ..., A_{ri}) = const. \; , \tag{7}$$

$$f(D_i^{-1}) \sim |D_i|^{-(p+1)/2} \; , \tag{8}$$

which correspondes to complete initial uncertainty about parameters A_{ij}, D_i, j=1,2,...,r_i, i=1,2,...,Q.

Theorem 1

Let the conditions A4 and A5 hold and let the multidimensional process $\{X_n\}$ is generated by the stochastic model (1), with parameters satisfying stationarity condition (2). Then, Bayesian estimate of the class conditional probability density function f($|\omega_i, Y_i$), for the given training set Y_i of length n(i), n(i)≥p+r_i +pr_i, has a form

$$f(X_1,\dots,X_N \mid \omega_i, Y_i) = f(X_1,\dots,X_r)\, \pi^{-(N-r_i)p/2} K(n(i), r_i, p)$$

$$\left| Y_{0i} Y_{0i}^T \right|^{p/2} \left| Y_{0i} Y_{0i}^T + \tilde{X}_0 \tilde{X}_0^T \right|^{-p/2} \left| S_i \right|^{-(N-r_i)/2} \cdot \tag{9}$$

$$\left| I_p + S_i^{-1} (\tilde{X}_1 - \hat{A}_i \tilde{X}_0)\, C_i\, (\tilde{X}_1 - \hat{A}_i \tilde{X}_0)^T \right|^{-[N+n(i)-2r_i-r_i p]/2} ,$$

$$K(n(i), r_i, p) = \prod_{k=1}^{p} \Gamma\left[\frac{N+n(i)+1-2r_i-r_i p-k}{2}\right] \prod_{k=1}^{p} \Gamma\left[\frac{n(i)+1-r_i-pr_i-k}{2}\right]^{-1}, \tag{10}$$

where Γ( ) denotes the Gamma function,

$$Y_{0i} = \begin{bmatrix} Y_{r_i}^{(i)} & Y_{r_i+1}^{(i)} & \dots & Y_{n(i)-1}^{(i)} \\ Y_{r_i-1}^{(i)} & Y_{r_i}^{(i)} & \dots & Y_{n(i)-2}^{(i)} \\ \dots & \dots & \dots & \dots \\ Y_1^{(i)} & Y_2^{(i)} & \dots & Y_{n(i)-r_i} \end{bmatrix}, \ dim\ Y_{0i} = r_i p\,[n(i)-r_i] , \tag{11}$$

$$Y_{1i} = \begin{bmatrix} Y_{r_i+1}^{(i)} & Y_{r_i+2}^{(i)} & \cdots & Y_{n(i)}^{(i)} \end{bmatrix}, \ dim\ Y_{1i} = p\,[n(i)-r_i] , \tag{12}$$

$$\tilde{X}_0 = \begin{bmatrix} X_{r_i} & X_{r_i+1} & \dots & X_{N-1} \\ X_{r_i-1} & X_{r_i} & \dots & X_{N-2} \\ \dots & \dots & \dots & \dots \\ X_1 & X_2 & \dots & X_{N-r_i} \end{bmatrix}, \ dim\ \tilde{X}_0 = r_i p(N-r) , \tag{13}$$

$$\tilde{X}_1 = \begin{bmatrix} X_{r_i+1} & X_{r_i+2} & \cdots & X_N \end{bmatrix}, \ dim\ \tilde{X}_1 = p(N-r_i) , \tag{14}$$

$$S_i = \left[Y_{1i} - \hat{A}_i Y_{0i} \right] \left[Y_{1i} - \hat{A}_i Y_{0i} \right]^T, \ dim\ S_i = p\ x\ p , \tag{15}$$

$$\hat{A}_i = \left[\hat{A}_{1i}\ \hat{A}_{2i}\ \cdots\ \hat{A}_{r_i i} \right] = Y_{1i} Y_{0i}^T \left[Y_{0i} Y_{0i}^T \right]^{-1}, dim\ \hat{A}_i = p\ x\ pr_i , \tag{16}$$

$$C_i = I_{N-r_i} - \tilde{X}_0^T \left[\tilde{X}_0 \tilde{X}_0^T + Y_{0i} Y_{0i} \right]^{-1} \tilde{X}_0 , \ dim\ C_i = (N-r_i)\ x\ (N-r_i). \tag{17}$$

Notice that matrix $\tilde{A}_i$ is a mean-square estimate of A_i, while the matrix S_i is proportional to the covariance matrix of the corresponding residuals, generated on the training sequence Y_i by the model with parameters $\hat{A}_i$.

II. REFERENCES

[1] M.Milosavljevic, "Analysis and Recognition of a Class of Multidimensional Signals", in Serbocroation, Ph.D. Thesis, School of Electrical Engineering, University of Belgrade, Yugoslavia 1982.

[2] L.D.Broemeling, "Bayesian Analysis of linear models", Marcel Dekker Inc., 1985.

[3] R.L.Kashyap, "Optimal Feature Selection and Decision Rules in Classification Problems with Time Series", IEEE Transactions on IT, vol.IT-24, No.3, May 1978.

[4] V.R.Marco, D.M.Yong, D.W.Turner, "Predictive Discrimination for Autoregressive Processes", Pattern Recognition Letters 7, 1988.

[5] M.Milosavljevic, "Optimal Decision Rule that Minimizes Conditional Probability of Error in Classification of Multidimensional Signals with Autoregressive Structure", in Serbocroation, Proceedings of the XXIX Conference ETAN, Nis, Yugoslavia, 1985.

EFFICIENT PARALLEL ALGORITHMS FOR STRING EDITING AND RELATED PROBLEMS

A. Apostolico[1] M.J. Atallah[2] L.L. Larmore[3] H.S. McFaddin[4]

Abstract. The string editing problem for input strings x and y consists of transforming x into y by performing a series of weighted edit operations on x of overall minimum cost. We give efficient PRAM parallel algorithms for this problem. If $m = \min(|x|, |y|)$ and $n = \max(|x|, |y|)$, then our CREW bound is $O(\log m \log n)$ time with $O(mn/\log m)$ processors. Our CRCW bound is $O((\log n(\log\log m)^2)$ time with $O(mn/\log\log m)$ processors. In either case, space is $O(mn)$.

1. Introduction

Let x be a string of $|x|$ symbols on some alphabet I. We consider three *edit operations* on x, namely, *deletion* of a symbol from x, *insertion* of a new symbol in x and *substitution* of one of the symbols of x with another symbol from I. Each edit operation has an associated nonnegative cost. Given two strings x and y, the *string editing problem* consists of finding a minimum cost sequence of edit operations that transforms x into y. This problem arises in many applications, and is solved by a serial algorithm in $\Theta(|x||y|)$ time and space [8].

Our main result is a CREW-PRAM algorithm for the string editing problem that runs in $O(\log m \log n)$ time with $O(mn/\log m)$ processors, where $m = \min\{|x|, |y|\}$ and $n = \max\{|x|, |y|\}$. We also give a CRCW-PRAM algorithm that runs in $O(\log n(\log\log m)^2)$ time with $O(mn/\log\log m)$ processors. In both algorithms, space is $O(mn)$. (Recall that the CREW-PRAM model allows concurrent reads but not concurrent writes, whereas the CRCW-PRAM allows both.)

In related work, Ranka and Sahni [7] have designed a hypercube algorithm for $m = n$ that runs in $O(\sqrt{n \log n})$ time with n^2 processors, and have considered time/processor tradeoffs. In independent work, Mathies [6] has obtained a CRCW-PRAM algorithm for the edit distance that runs in $O(\log n \log m)$ time with $O(mn)$ processors if the weight of every edit operation is smaller than a given constant integer. We have learned that Aggarwal and Park [1] have recently established bounds similar to ours.

Definition 1 *An $m \times n$ grid DAG is a directed acyclic graph whose vertices are the mn points of an $m \times n$ grid, and such that the only edges from grid point (i, j) are to grid points $(i, j+1)$, $(i+1, j)$ and $(i+1, j+1)$.*

Figure 1 shows an example of a grid DAG and also illustrates our convention of drawing the points such that point (i, j) is at the i-th row from the top and j-th column from the left. Note that the top-left point is $(1, 1)$ and has no edge coming into it (i.e. is a *source*), and that the bottom-right point is (m, n) has no edge leaving it (i.e. is a *sink*).

From the known sequential solution to the string editing problem, it easily follows that it reduces to a path problem on an $(|x| + 1) \times (|y| + 1)$ grid DAG G: that of computing a shortest source-to-sink path in G. We henceforth only concern ourselves with the latter path problem. Although we focus on the problem of computing the *length* (i.e., cost) of the optimal path, our procedures can easily be used to compute the path itself (we give the details of the modifications needed in [2]).

Throughout, the *left boundary* of G is the set of points in its leftmost column. The *right*, *top*, and *bottom* boundaries are analogously defined. The *boundary* of G is the union of its left, right, top and bottom boundaries.

---

[1] Purdue U. and L'Aquila. Supported in part by the Italian Ministry of Education, National Research Council through IASI-CNR, and the British Research Council under SERC Grant GR-E76797.

[2] Purdue U. Supported by ONR Grants N00014-84-K-0502 and N00014-86-K-0689, NSF Grant DCR-8451393, with matching funds from AT&T. Part of this work was done at RIACS, NASA Ames Res. Center.

[3] UC Irvine. Supported by UC Irvine.

[4] Purdue U. Research carried out at RIACS, NASA Ames Research Center.

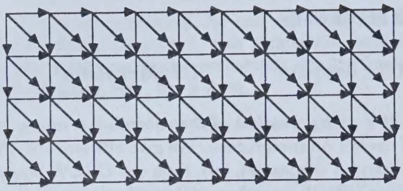


Fig. 1. A 5 × 10 grid DAG.

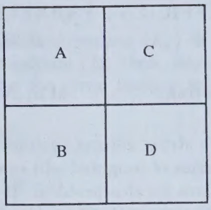


Fig. 2. Partitioning the problem.

Section 2 gives a preliminary CREW-PRAM algorithm for the case $m = n$. Section 3 gives an algorithm that uses a factor of $\log m$ fewer processors than the previous one and that will be later needed in our best CREW algorithm (given in Section 5). Section 4 discusses the case $m \leq n$. Section 5 gives our best CREW-PRAM algorithm. Section 6 gives the CRCW-PRAM algorithm.

2. A preliminary algorithm

Let G be an $m \times m$ grid DAG, and let $DIST_G$ be the $(2m) \times (2m)$ matrix containing the lengths of all shortest paths that begin at the top or left boundary of G, and end at the right or bottom boundary of G. In this section we establish that the matrix $DIST_G$ can be computed in $O(\log^3 m)$ time, $O(m^2)$ space, and with $O(m^2/\log m)$ processors by a CREW-PRAM. The preliminary algorithm that achieves this is intended as a "warmup" for the better algorithms that follow in later sections. It works as follows: divide the $m \times m$ grid into four $(m/2) \times (m/2)$ grids A, B, C, D, as shown in Figure 2. In parallel, recursively solve the problem for each of the four grids A, B, C, D, obtaining the four distance matrices $DIST_A$, $DIST_B$, $DIST_C$, $DIST_D$. Then obtain from these four matrices the desired matrix $DIST_G$. The main problem is how to perform this last step efficiently. The performance bounds we claimed for this preliminary algorithm would immediately follow if we can show that, for any integer $q \leq m$ of our choice, $DIST_G$ can be obtained from $DIST_A$, $DIST_B$, $DIST_C$, $DIST_D$ in time $O((q + \log m) \log m)$ and with $O(m^2/q)$ processors. This is because the time and processor complexities of the overall algorithm would then obey the recurrences $T(m) \leq T(m/2) + c_1(q + \log m)\log m$, and $P(m) \leq \max(4P(m/2), c_2 m^2/q)$, whose solutions are $T(m) = O((q + \log m)\log^2 m)$ and $P(m) = O(m^2/q)$. Choosing $q = \log m$ would then establish the desired result. Therefore in the rest of this section, we merely concern ourselves with showing that $DIST_G$ can be obtained from $DIST_A$, $DIST_B$, $DIST_C$, $DIST_D$ in time $O((q + \log m)\log m)$ and with $O(m^2/q)$ processors.

Let $DIST_{A \cup B}$ be the $(3m/2) \times (3m/2)$ matrix containing the lengths of shortest paths that begin on the top or left boundary of $A \cup B$ and end on its right or bottom boundary. Let $DIST_{C \cup D}$ be analogously defined for $C \cup D$. The procedure for obtaining $DIST_G$ performs the following Steps 1–3: (1) Use $DIST_A$ and $DIST_B$ to obtain $DIST_{A \cup B}$, (2) Use $DIST_C$ and $DIST_D$ to obtain $DIST_{C \cup D}$, and (3) Use $DIST_{A \cup B}$ and $DIST_{C \cup D}$ to obtain $DIST_G$. We only show how Step (1) is done, since the procedures for Steps (2) and (3) are very similar. Clearly, we need only compute the entries of $DIST_{A \cup B}$ that correspond to paths that begin on the top or left boundary of A and end on the right or bottom boundary of B. Assign to every point v on the top or left boundary of A a group of m/q processors whose task is to compute the lengths of all shortest paths that begin at v and end on the right or bottom boundary of B. It suffices to show that it can indeed do this in time $O((q + \log m)\log m)$. Observe that:

$$DIST_{A \cup B}(v, w) = \min\{DIST_A(v, p) + DIST_B(p, w) \mid p \text{ is on the boundary common to } A \text{ and } B\} \quad (1)$$

Definition 2 *Let v be any point on the left or top boundary of A, and let w be any point*

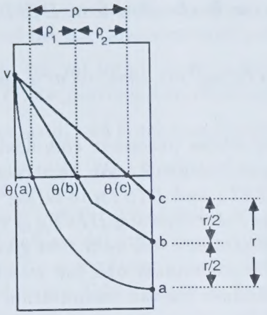


Fig. 3. Illustrating the preliminary algorithm.

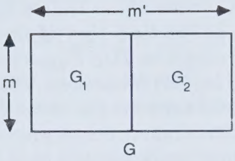


Fig. 4. Illustrating Lemma 2.

on the bottom or right boundary of B. Let $\theta(v, w)$ denote the leftmost *p which minimizes the right-hand-side of (1). Equivalently, $\theta(v, w)$ is the leftmost point of the common boundary of A and B such that a shortest v-to-w path goes through it.*

Define a linear ordering $<_B$ on the m points at the bottom and right boundaries of B, such that they are encountered in increasing order of $<_B$ by a walk that starts at the leftmost point of the lower boundary of B and ends at the top of the right boundary of B. Let L_B be the list of m points on the lower and right boundaries of B, sorted by increasing order according to the $<_B$ relationship. For any $w_1, w_2 \in L_B$, we have the following:

$$\text{If } w_1 <_B w_2 \text{ then } \theta(v, w_1) \text{ is not to the right of } \theta(v, w_2). \tag{2}$$

Before proving property (2), we sketch how it is used to obtain an $O((q + \log(m/q)) \log m)$ time and $O(m/q)$ processor algorithm for computing $DIST_{A \cup B}(v, w)$ for all $w \in L_B$. For the rest of this section, we use $\theta(w)$ as a shorthand for $\theta(v, w)$, with v being understood. It suffices to compute $\theta(w)$ for all $w \in L_B$. The procedure for doing this is recursive, and takes as input three things: (i) a particular range of r contiguous values in L_B, say a range that begins at point a and ends at point c, $a <_B c$, (ii) the points $\theta(a)$ and $\theta(c)$, and (iii) a number of processors equal to $\max\{1, (\rho + r)/q\}$ where ρ is the number of points between $\theta(a)$ and $\theta(c)$ on the boundary common to A and B. (See Figure 3.) The procedure returns $\theta(w)$ for every $a <_B w <_B c$. If $r = 1$ then there is only one such w and there are enough processors to compute $\theta(w)$ in time $O(q + \log(\rho/q))$. If $r > 1$ then all of the $\max\{1, (\rho + r)/q\}$ processors get assigned to the median of the a-to-c range and compute, for that median (call it point b), the value $\theta(b)$ in time $O(q + \log(\rho/q))$. Because of (2), it is now enough for the procedure to recursively call itself on the a-to-b range and (in parallel) the b-to-c range. The first (resp. second) of these recursive calls gets assigned $\max\{1, (\rho_1 + r/2)/q\}$ (resp. $\max\{1, (\rho_2 + r/2)/q\}$) processors, where ρ_1 (resp. ρ_2) is the number of points between $\theta(a)$ and $\theta(b)$ (resp. between $\theta(b)$ and $\theta(c)$). Because $\rho_1 + \rho_2 = \rho$, there are enough processors available for the two recursive calls. (See Figure 3.) In the initial call to the procedure, it is given (i) the whole list L_B, (ii) the θ of the first and last point of L_B, and (iii) $3m/2q$ processors. The depth of the recursion is $\log m$, at each level of which the time taken is no more than $O(q + \log(m/q))$. Therefore the procedure takes time $O((q + \log(m/q)) \log m)$ with $O(m/q)$ processors. We conclude that the preliminary solution would immediately follow if we establish property (2). We refer the reader to [2] for a proof of (2).

3. Using fewer processors

This section gives an algorithm that has same time complexity as that of the previous section, but whose processor complexity is a factor of $\log m$ better. This is more than a mere "warmup" for our best CREW algorithm of Section 5: the algorithm of Section 5 will actually use the technical result, given in this section, that $DIST_{A \cup B}$ can be obtained from $DIST_A$ and $DIST_B$ with $O(m^2)$ total work.

Lemma 1 *The matrix $DIST_G$ can be computed in $O(\log^3 m)$ time, $O(m^2)$ space, and with $O(m^2/\log^2 m)$ processors by a CREW-PRAM.*

We prove the above lemma by giving an algorithm whose processor complexity is a $\log m$ factor better than that of the preliminary solution of Section 2. We illustrate the method by showing how $DIST_{A \cup B}$ can be obtained from $DIST_A$ and $DIST_B$ in $O(\log^2 m)$ time and $O(m^2/\log^2 m)$ processors. The preliminary procedure for computing $DIST_{A \cup B}$ can be seen to do a total amount of work which is $O(m^2 \log m)$. Our strategy will be to first give a procedure which has same time and processor complexities as the preliminary one, but which does a total amount of work which is only $O(m^2)$. Our claimed bounds for the computation of $DIST_{A \cup B}$ from $DIST_A$ and $DIST_B$ will then immediately follow from this improved procedure, by Brent's theorem [4].

Let L_A and $<_A$ be defined analogously to L_B and $<_B$, respectively. In other words, L_A is a list of the m points on the left and top boundaries of A, sorted in the order in which they are encountered by a walk that starts at the lowest point of the left boundary of A and ends at the rightmost point of the top boundary of A (i.e. sorted by increasing order according to the $<_A$ relationship). A symmetric version of (2) holds, i.e., for any $w \in L_B$ and any two points v_1 and v_2 of L_A, we have the following:

$$\text{If } v_1 <_A v_2 \text{ then } \theta(v_1, w) \text{ is not to the right of } \theta(v_2, w). \tag{3}$$

Let P be the $m \times (m/2)$ submatrix of $DIST_A$ containing the lengths of the shortest paths that begin at the top or left boundary of A, and end at its bottom boundary. Let Q be the $(m/2) \times m$ submatrix of $DIST_B$ containing the lengths of the shortest paths that begin at the top boundary of B, and end at its bottom or right boundary. By definition, the rows of P are indexed by the entries of L_A, the columns of Q are indexed by the entries of L_B, and the columns of P (hence the rows of Q) are indexed by the $m/2$ points at the common boundary of A and B, sorted from left to right. The problem we face is that of "multiplying" the $m \times (m/2)$ matrix P and the $(m/2) \times m$ matrix Q in the closed semiring $(\min, +)$. In matrix terminology, $\theta(v, w)$ is the smallest index k, $1 \leq k \leq m/2$, such that $PQ(v, w) = P(v, k) + Q(k, w)$. We give the procedure below for the (more general) case where P is an $\ell \times h$ matrix, and Q is an $h \times \ell$ matrix, $\ell \leq 2h$. The only structure of these matrices that our algorithm uses is the following property (4), which is merely a re-statement of properties (2) and (3) using matrix terminology:

$$\forall (1 \leq v_1 < v_2 \leq \ell, 1 \leq w \leq \ell), \theta(v_1, w) \leq \theta(v_2, w), \text{ and } \theta(w, v_1) \leq \theta(w, v_2). \tag{4}$$

To compute the product of P and Q in the closed semiring $(\min, +)$, it suffices to compute $\theta(v, w)$ for all $1 \leq v, w \leq \ell$. To compute the product PQ (i.e. the function θ), we use the following procedure which runs in $O(\log \ell \log h)$ time and $O(\ell h/\log h)$ processors and $O(\ell h)$ total work:

1. Recursively solve the problem for the product $P'Q'$ where P' (resp. Q') is the $(\ell/2) \times h$ (resp. $h \times (\ell/2)$) matrix consisting of the odd rows (resp. odd columns) of P (resp. Q). This gives $\theta(v, w)$ for all pairs (v, w) whose respective parities are (odd,odd). If $W(\ell, h)$ and $T(\ell, h)$ denote the total work and time for this procedure, then this step does $W(\ell/2, h)$ work in $T(\ell/2, h)$ time.

2. Compute $\theta(v, w)$ for all pairs (v, w) of parities (even,odd). This is done as follows. In parallel for each odd w, assign $h/\log h$ processors to w, with the task of computing $\theta(v, w)$ for all even v. The fact that we already know $\theta(v, w)$ for all odd v, together with property (4), implies that these $h/\log h$ processors are enough to do the job in $O(\log h)$ time. The work done is then $O(h)$ for each such w, for a total of $O(\ell h)$ work for this step.

3. Compute $\theta(v,w)$ for all pairs (v,w) of parities (odd,even). The method used is identical to that of the previous step and is therefore omitted.
4. Compute $\theta(v,w)$ for all pairs (v,w) of parities (even,even). The method is very similar to that of the previous two steps and is therefore omitted.

The time, processor, and work complexities of the above method satisfy the recurrences:

$$T(\ell,h) \leq T(\ell/2,h)+c_1 \log h,\ P(\ell,h) \leq \max\{P(\ell/2,h),\ell h/\log h\},\ W(\ell,h) \leq W(\ell/2,h)+c_2\ell h,$$

whose solutions are $T(\ell,h) = O(\log \ell \log h)$, $P(\ell,h) = O(\ell h/\log h)$, and $W(\ell,h) = O(\ell h)$. By Brent's Theorem [4], this implies that the above algorithm can be simulated by $\ell h/q$ processors in $O(q + \log \ell \log h)$ time. In our case, we have $\ell = m$ and $h = m/2$, implying that PQ (and hence $DIST_{A\cup B}$) can be obtained from P and Q in $O(q + \log^2 m)$ time with $O(m^2/q)$ processors.

The above method enables us to obtain $DIST_G$ from $DIST_A, DIST_B, DIST_C, DIST_D$ in $O(q+\log^2 m)$ time and $O(m^2/q)$ processors. This implies that the overall divide-and-conquer algorithm runs in $O((q + \log^2 m)\log m)$ time with $O(m^2/q)$ processors. Choosing $q = \log^2 m$ establishes Lemma 1.

4. The case $m \leq n$

We prove the following.

Theorem 1 *If G is $m \times n$, $m \leq n$, then the length of a shortest source-to-sink path can be computed by a CREW-PRAM in $O(\log n \log^2 m)$ time, $O(mn)$ space, and with $O(mn/\log^2 m)$ processors.*

First, observe that we can no longer afford to compute the $DIST_G$ matrix, since it is $(m + n) \times (m + n)$. Instead, we compute an $m \times m$ matrix D_G that contains the lengths of all the shortest paths that begin at the left boundary of G, and end at the right boundary of G.

Lemma 2 *Let G be an $m \times m'$ grid DAG that is partitioned by a vertical line into G_1 and G_2. (See Figure 4.) Then, given D_{G_1} and D_{G_2}, the matrix D_G can be computed by a CREW-PRAM in $O(\log^2 m)$ time, $O(m^2)$ space, and with $O(m^2/\log^2 m)$ processors.*

Proof. Similar to the procedure of Section 3 for computing $DIST_{A\cup B}$, and omitted. □

Proof of Theorem 1. Without loss of generality, assume that m divides n (if not then G can always be "padded" with extra vertices and zero-cost edges so as to make it $m \times n'$ where m divides n' and $n' - n \leq m$). Partition G by vertical lines into n/m grid DAGs $G_1, \ldots, G_{n/m}$, where each G_i is $m \times m$ (see Figure 5). In parallel for each $i \in \{1, \ldots, n/m\}$, use Lemma 1 to obtain the $DIST_{G_i}$ matrices. This takes $O(\log^3 m)$ time with a total of $O((m^2/\log^2 m)(n/m)) = O(mn/\log^2 m)$ processors. From each $DIST_{G_i}$ matrix, extract its submatrix D_{G_i}. We are now left with the task of combining the D_{G_i}'s into a single D_G. In parallel, we recursively obtain the D-matrix of the union of the leftmost $n/2m$ G_i's, and similarly the D-matrix of the union of the rightmost $n/2m$ G_i's. We then combine these two D matrices into D_G by using Lemma 2. This recursive combining procedure takes a total of $O(\log^2 m \log(n/m))$ time with $O(mn/\log^2 m)$ processors. The overall time complexity is therefore $O(\log^3 m + \log^2 m \log(n/m)) = O(\log n \log^2 m)$. □

5. A faster CREW-PRAM algorithm

This section gives our best CREW algorithm:

Theorem 2 *Let G be an $m \times n$ grid DAG, $m \leq n$. A shortest source-to-sink path in G can be computed by a CREW-PRAM in $O(\log n \log m)$ time, $O(mn)$ space, and with $O(mn/\log m)$ processors.*

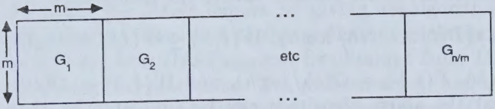


Fig. 5. Partitioning G.

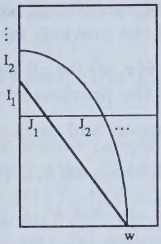


Fig. 6. The second stage.

From the developments of Sections 2–4, it should be clear that in order to establish the above theorem, it suffices to show the following: (1) that the matrix $DIST_{A\cup B}$ can be obtained from $DIST_A$ and $DIST_B$ in $O(\log m)$ time, $O(m^2)$ space, and with $O(m^2/\log m)$ processors, and (2) that the matrix D_G can be obtained from D_{G_1} and D_{G_2} (see Figure 4) in $O(\log m)$ time, $O(m^2)$ space, and with $O(m^2/\log m)$ processors. The rest of this section proves (1) (the proof for (2) is very similar).

5.1. Obtaining one row of $DIST_{A\cup B}$

This subsection gives an $O(\log m)$ time, $O(m \log m)$ space, and $O(m \log m)$ processor algorithm for obtaining one particular row of $DIST_{A\cup B}$, i.e. computing $\theta(v, w)$ for a fixed $v \in L_A$ and all $w \in L_B$. The fixed vertex v is implicit in the rest of this subsection, so that whenever we refer to a "path to w" it is understood that this path originates at v.

We refer to the vertices on the boundary common to A and B (denoted $A \cap B$ for short) as *crossing vertices* and number them $c_1, c_2, \ldots, c_{m/2}$, where the numbering is from left to right along the common boundary. We refer to the vertices in L_B as *destination vertices* and denote them w_1, w_2, $\ldots$, w_m, numbered according to $<_B$, their order in L_B.

Definition 3 *A* crossing interval *is a non-empty set of contiguous crossing vertices* $\{c_i, c_{i+1}, \ldots, c_j\}$. *A crossing interval I is* to the left of *another crossing interval J if the rightmost vertex of I is to the left of the leftmost vertex of J.*

Definition 4 *Let $F \subseteq A \cap B$ and $w \in L_B$, i.e. F is a set of crossing vertices (not necessarily an interval) and w is a destination vertex. Let $\theta_F(w)$ denote the leftmost crossing vertex in F incident to a (v, w) path that is shortest among all (v, w) paths constrained to pass through F.*

Note that $\theta_F(w)$ may differ from $\theta(v, w)$, but that $\theta_{A\cap B}(w) = \theta(v, w)$. The following lemma is the analogue, for constrained paths, to property (2) of Section 2.

Lemma 3 *Let $F \subseteq A \cap B$ and $w_1, w_2 \in L_B$. If $w_1 <_B w_2$, then $\theta_F(w_1)$ is not to the right of $\theta_F(w_2)$.*

Proof. Identical to that of property (2) of Section 2, and omitted. □

We now give an informal description of the algorithm. If U is any set of destination vertices and I is any crossing interval, then we will define $\theta_I(U)$ to be a data structure that contains enough information to determine $\theta_I(w)$ for all $w \in U$. The details of that data structure will be explained later. Also, it is useful to think of the computation as progressing through the nodes of a tree T which we now proceed to define.

We define a crossing interval to be *diadic* if it is either $A \cap B$ (i.e. it consists of all crossing vertices), or if it is the the left or right half of a diadic crossing interval. Note that there are exactly $m - 1$ diadic crossing intervals, which form a complete binary tree T rooted at $A \cap B$, and whose $m/2$ leaves are the $m/2$ crossing vertices (the i-th leaf of T containing c_i, the i-th leftmost crossing vertex). Thus the diadic crossing interval at an interior node of T is simply the union of the diadic crossing intervals of its two children in T. We can talk about the *height* and the *children* of a diadic crossing interval (= its height and children in T).

Since the $m-1$ diadic crossing intervals are the only crossing intervals we shall be interested in, from now on we simply say "interval" as a shorthand for "diadic crossing interval". Thus whenever we refer to an interval I we are implicitly assuming that $I \in T$, i.e. that I is one of the $m-1$ diadic crossing intervals. We use $|I|$ to denote the size of the interval, i.e. the number of crossing vertices in it. Observe that $\sum_{I \in T} |I| = O(m \log m)$. Thus we have enough processors to associate $|I|$ of them with each interval I (i.e. node I) of T. Similarly, we can afford to use $O(|I|)$ space per interval I. The computation proceeds in $2 \log m - 1$ *stages*, each of which takes constant time. The ultimate goal is for every interval I to compute $\theta_I(L_B)$. The structure of the algorithm is reminescent of the *cascading divide-and-conquer* technique [5,3]: each $I \in T$ will compute $\theta_I(U)$ for progressively larger subsets U of L_B, subsets U that double in size from one stage to the next of the computation. We now proceed to state precisely what these subsets are.

Definition 5 *A* k-sample *of* L_B *is obtained by choosing every* k*-th element of* L_B *(i.e. every element whose rank in* L_B *is a multiple of* k*). For example, a 4-sample of* L_B *is* $(w_4, w_8,, w_12, \ldots)$. *For* $k \in \{0, 1, \ldots, \log m\}$, *let* U_k *denote an* $(m/2^k)$*-sample of* L_B.

For example:
$U_0 = \{w_m\}$,
$U_1 = \{w_{m/2}, w_m\}$,
$U_2 = \{w_{m/4}, w_{m/2}, w_{3m/4}, w_m\}$,
$U_3 = \{w_{m/8}, w_{m/4}, w_{3m/8}, w_{m/2}, w_{5m/8}, w_{3m/4}, w_{7m/8}, w_m\}$,
...
$U_{\log m} = \{w_1, w_2, \ldots, w_m\} = L_B$.

Note that $|U_k| = 2^k = 2|U_{k-1}|$.

At the t-th stage of the algorithm, an interval I of height h in T will use its $|I|$ processors to compute, in constant time, $\theta_I(U_{t-h})$ if $h \le t \le h + \log m$. It does so with the help of information from $\theta_I(U_{t-1-h})$, $\theta_{LeftChild(I)}(U_{t-h})$, and $\theta_{RightChild(I)}(U_{t-h})$, all of which are available from the previous stage $t-1$. If $h > t$ or $t > h + \log m$ then interval I does nothing during stage t. Thus before stage h the interval I lies "dormant", then at stage $t = h$ it first "wakes up" and computes $\theta_I(U_0)$, then at the next stage $t = h + 1$ it computes $\theta_I(U_1)$, etc. At step $t = h + \log m$ it computes $\theta_I(U_{\log m})$, after which it is done. The details of what information I stores and how it uses its $|I|$ processors to perform stage t in constant time are given below. First, we observe the following.

Lemma 4 *The algorithm terminates after* $2 \log m - 1$ *stages.*

Proof. After stage $h + \log m$ every interval I of height h is done, i.e. it has computed $\theta_I(L_B)$. The root interval has height $\log m - 1$ and thus is done after stage $2 \log m - 1$. □

Thus to establish the main claim of this subsection, it suffices to prove the following lemma.

Lemma 5 *With* $|I|$ *processors and* $O(|I|)$ *space assigned to each interval* $I \in T$, *every stage of the algorithm can be completed in constant time.*

The rest of this subsection proves the above lemma. We begin by describing the way in which an interval I at height h in T stores $\theta_I(U_{t-h})$ using only $|I|$ space. Rather than directly storing the values $\theta_I(w)$ for all $w \in U_{t-h}$ (which would require $|U_{t-h}|$ space), we store instead the *inverse* mapping, which turns out to have a compact $O(|I|)$ space encoding because of the monotonicity property guaranteed by Lemma 3. In other words, for each $c \in I$, let $\pi_I(c, t) = \{w \in U_{t-h} | \theta_I(w) = c\}$. Then Lemma 3 implies that the elements of $\pi_I(c, t)$ are contiguous in the list U_{t-h}. More specifically, the sets $\pi_I(c, t)$, $c \in I$, form a partition of the set U_{t-h} into $|I|$ subsets each of which is either empty or contains contiguous elements in U_{t-h}. Therefore I does not need to store the elements of $\pi_I(c, t)$ explicitly, but rather by just remembering where they begin and end in U_{t-h}, i.e. $O(1)$ space for each $c \in I$. Of course U_{t-h} is itself not stored explicitly by I, since the height h and stage number t implicitly determine it. Thus $O(|I|)$ space is enough for storing $\pi_I(c,t)$ for all $c \in I$.

Interval I stores the sets $\pi_I(c,t)$, $c \in I$, in an array $RANGE_I$, with entries $RANGE_I(c) = (w_i, w_j)$ such that w_i (resp. w_j) is the first (resp. last) element of U_{t-h} that belongs to $\pi_I(c,t)$. If $\pi_I(c,t)$ is empty then $RANGE_I(c)$ equals $\emptyset$. At stage t of the algorithm, I must update the $RANGE_I$ array so that it changes from being a description of the $\pi_I(c,t-1)$'s to being a description of the $\pi_I(c,t)$'s. The rest of this subsection need only show how such an update is done in constant time by the $|I|$ processors assigned to I. Of course, since we are ultimately interested in $\theta_{A\cap B}(w)$ for every $w \in L_B$, at the end of the algorithm we must run a postprocessing procedure which recovers this information from the $RANGE_{A\cap B}$ array available at the root of T, i.e. it explicitly obtains $\theta_{A\cap B}(w)$ for all $w \in U_{\log m}$. But this postprocessing is trivial to perform in $O(\log m)$ time with $O(m)$ processors, and we shall not concern ourselves with it any more.

In the rest of this subsection, intervals L and R are the left and (respectively) right children of I in T. Observe that, for any destination w, $\theta_I(w)$ is one of $\theta_L(w)$ or $\theta_R(w)$. Furthermore, if $\theta_I(w) = \theta_L(w)$ then $\theta_I(w') \in L$ for every w' smaller than w (in the $<_B$ ordering). Similarly, if $\theta_I(w) = \theta_R(w)$ then $\theta_I(w') \in R$ for any w' larger than w. (These observations follow from Lemma 3.)

The $RANGE_I$ array alone is not enough to enable I to perform the updating required at stage t. In addition, at each stage t, I must compute in a register called $CRITICAL_I$ an entry $Critical_I(t)$ defined as follows.

Definition 6 *At each stage t, let the* critical destination *for I, denoted $Critical_I(t)$, be the largest $w \in U_{t-h}$ such that $\theta_I(w) = \theta_L(w)$. If there is no such w (i.e. if $\theta_I(w) = \theta_R(w)$ for all $w \in U_{t-h}$), then $Critical_I(t) = \emptyset$.*

Note that Lemma 3 ensures that $Critical_I(t)$ is well defined. We shall later show how storing and maintaining this critical destination enables I to update the $RANGE_I$ array in constant time. Of course it also places on I the burden of updating its $CRITICAL_I$ register so that after stage t it contains $Critical_I(t)$ rather than $Critical_I(t-1)$. We shall later show that updating the $CRITICAL_I$ register can be done in constant time as well.

We now complete this subsection by explaining how I performs stage t, i.e. how it obtains $Critical_I(t)$ and the $\pi_I(c,t)$'s using the $\pi_L(c,t-1)$'s, the $\pi_R(c,t-1)$'s, and its previous critical index $Critical_I(t-1)$. We need the following three observations (whose correctness follows from the definitions):
$Critical_I(t)$ is either $Critical_I(t-1)$, or the successor of $Critical_I(t-1)$ in U_{t-h}. (5)
If $c \in L$ then $\pi_I(c,t) = \pi_L(c,t-1) - \{$the elements of $\pi_L(c,t-1)$ that are larger than $Critical_I(t)$ in the $<_B$ ordering$\}$. (6)
If $c \in R$ then $\pi_I(c,t) = \pi_R(c,t-1) - \{$the elements of $\pi_R(c,t-1)$ that are less than or equal to $Critical_I(t)$ in the $<_B$ ordering$\}$. (7)

Updating the $CRITICAL_I$ register

Relationship (5) implies that in order to update $CRITICAL_I$ (i.e. compute $Critical_I(t)$) all I has to do is determine which of $Critical_I(t-1)$ or its successor in U_{t-h} is the correct value of $Critical_I(t)$. This is done as follows. If $Critical_I(t-1)$ has no successor in U_{t-h} then $Critical_I(t-1) = w_m$ and hence $Critical_I(t) = Critical_I(t-1)$. Otherwise the updating is done in the following two steps. For shorthand, let r denote $Critical_I(t-1)$, and let s denote the successor of r in U_{t-h}.

- The first step is to compute $\theta_L(s)$ and $\theta_R(s)$ in constant time. This involves a search in L (resp. R) for the crossover c in L (resp. R) whose $\pi_L(c,t-1)$ (resp. $\pi_R(c,t-1)$) contains s. These two searches in L and R are done in constant time with the $|I|$ processors available. We explain how the search in L is done (that in R is similar and omitted). I assigns a processor to each $c \in L$, and that processor tests whether s is in $\pi_L(c,t-1)$;

the answer is "yes" for exactly one of those $|L|$ processors and thus can be collected in constant time. Thus I can determine $\theta_L(s)$ and $\theta_R(s)$ in constant time.

- The next step consists of comparing which of the following two paths to s is better: the one through $\theta_L(s)$, or the one through $\theta_R(s)$. If the path through $\theta_R(s)$ is the better of the two then $Critical_I(t)$ is the same as $Critical_I(t-1)$ and the $CRITICAL_I$ register stays the same (containing r). Otherwise $Critical_I(t)$ is s, and we set $CRITICAL_I$ equal to s. This comparison of the two paths and resulting update are done in constant time (by one processor, in fact).

We next show how the just computed $Critical_I(t)$ value is used to compute the $\pi_I(c,t)$'s in constant time.

Updating the $RANGE_I$ array

Relationship (6) implies the following for each $c \in L$:

1. If $\pi_L(c,t-1)$ is to the left of $Critical_I(t)$ then $\pi_I(c,t) = \pi_L(c,t-1)$.
2. If $\pi_L(c,t-1)$ is to the right of $Critical_I(t)$ then $\pi_I(c,t) = \emptyset$.
3. If $\pi_L(c,t-1)$ contains $Critical_I(t)$ then it consists of the portion of $\pi_L(c,t-1)$ up to (and including) $Critical_I(t)$.

The above facts 1–3 immediately imply that $O(1)$ time is enough for $|L|$ of the $|I|$ processors assigned to I to compute $\pi_I(c,t)$ for all $c \in L$, by adjusting the $RANGE_I(c)$ value according to rules 1–3 above (recall that the $\pi_L(c,t-1)$'s are available in L from the previous stage $t-1$, and $Critical_I(t)$ has already been computed and is in the $CRITICAL_I$ register).

A similar argument shows that relationship (7) implies that $|R|$ processors are enough for computing $\pi_I(c,t)$ for all $c \in R$. Thus I can update its $RANGE_I$ array in constant time with $|I|$ processors. This completes the proof of Lemma 5.

5.2. Obtaining all rows of $DIST_{A \cup B}$

This subsection shows that $O(m^2/\log m)$ processors and $O(m^2)$ space suffice for computing in $O(\log m)$ time all the $\theta(v,w)$'s (hence for computing the $DIST_{A\cup B}$ matrix). Let L_A and L_B be as in previous sections. Our task is to compute $\theta(v,w)$ for all $v \in L_A$ and all $w \in L_B$. We use $S(L,k)$ to denote the k-sample of a list L.

In the first stage of the computation, we assign $m \log m$ processors to each $v \in S(L_A, \log^2 m)$. Then, in parallel for all $v \in S(L_A, \log^2 m)$, we use the method of the previous subsection to obtain $\theta(v,w)$ for all $w \in L_B$. This first stage of the computation takes $O(\log m)$ time, $O(m^2)$ space, and $O(m^2/\log m)$ processors, and obtains $\theta(v,w)$ for all $v \in S(L_A, \log^2 m)$ and $w \in L_B$.

In the second stage of the computation, we assign $2m$ processors to each $w \in S(L_B, \log m)$, with the task of computing $\theta(v,w)$ for all $v \in L_A$. These $2m$ processors perform this computation for their particular w in $O(\log m)$ time, as follows. The set of $m/\log^2 m$ values $\{\theta(v,w) \mid v \in S(L_A, \log^2 m)\}$ partitions the common boundary of A and B into $m/\log^2 m$ pieces $J_1, J_2, \ldots$ (see Figure 6). Let $I_1, I_2, \ldots$ be the $m/\log^2 m$ pieces (of size $\log^2 m$ each) into which $S(L_A, \log^2 m)$ partitions L_A (see Figure 6). Partition the group of $2m$ processors assigned to w into $m/\log^2 m$ subgroups, where the i-th subgroup contains $\log^2 m + |J_i|$ processors whose task is to compute, for all $v \in I_i$, which element of J_i equals $\theta(v,w)$. This subgroup of $\log^2 m + |J_i|$ processors does this as follows.

1. It gives each of the $\log m$ elements of $S(I_i, \log m)$ (say, to element v) $1 + |J_i|/\log m$ processors that v uses to find out, in $O(\log m)$ time, which element of J_i equals $\theta(v,w)$. The set of $\log m$ values $\{\theta(v,w) \mid v \in S(I_i, \log m)\}$ partitions J_i into $\log m$ pieces $J_{i,1}, J_{i,2}, \ldots$. Let $I_{i,1}, I_{i,2}, \ldots$ be the $\log m$ pieces (of size $\log m$ each) into which $S(I_i, \log m)$ partitions I_i.

2. It partitions its $\log^2 m + |J_i|$ processors into $\log m$ subsubgroups, where the k-th subsubgroup contains $\log m + |J_{i,k}|$ processors whose task is to compute, for all $v \in I_{i,k}$, which element of $J_{i,k}$ equals $\theta(v, w)$. This subsubgroup of $\log m + |J_{i,k}|$ processors does this in $O(\log m)$ time by giving to each of the $\log m$ elements of $I_{i,k}$ (say, to element v) $1 + |J_{i,k}|/\log m$ processors that v uses to find out, in $O(\log m)$ time, which element of $J_{i,k}$ equals $\theta(v, w)$.

In the third stage of the computation, we assign $2m/\sqrt{\log m}$ processors to each $v \in S(L_A, \sqrt{\log m})$, with the task of computing $\theta(v, w)$ for all $w \in L_B$. These $2m/\sqrt{\log m}$ processors perform this computation for their particular v in $O(\log m)$ time, as follows. The set of $m/\log m$ values $\{\theta(v, w) \mid w \in S(L_B, \log m)\}$ partitions the common boundary of A and B into $m/\log m$ pieces $J_1, J_2, \ldots$. Let $I_1, I_2, \ldots$ be the $m/\log m$ pieces (of size $\log m$ each) into which $S(L_B, \log m)$ partitions L_B. Partition the group of $2m/\sqrt{\log m}$ processors assigned to v into $m/\log m$ subgroups, where the i-th subgroup contains $\sqrt{\log m} + |J_i|/\sqrt{\log m}$ processors whose task is to compute, for all $w \in I_i$, which element of J_i equals $\theta(v, w)$. This subgroup of $\sqrt{\log m} + |J_i|/\sqrt{\log m}$ processors does this as follows.

1. It gives each of the $\sqrt{\log m}$ elements of $S(I_i, \sqrt{\log m})$ (say, to element w) $1 + |J_i|/\log m$ processors that w uses to find out, in $O(\log m)$ time, which element of J_i equals $\theta(v, w)$. The set of $\sqrt{\log m}$ values $\{\theta(v, w) \mid w \in S(I_i, \sqrt{\log m})\}$ partitions J_i into $\sqrt{\log m}$ pieces $J_{i,1}, J_{i,2}, \ldots$. Let $I_{i,1}, I_{i,2}, \ldots$ be the $\sqrt{\log m}$ pieces (of size $\sqrt{\log m}$ each) into which $S(I_i, \sqrt{\log m})$ partitions I_i.
2. It partitions its $\sqrt{\log m} + |J_i|/\sqrt{\log m}$ processors into $\sqrt{\log m}$ subsubgroups. The k-th subsubgroup contains $1 + |J_{i,k}|/\sqrt{\log m}$ processors whose task is to compute, for all $w \in I_{i,k}$, which element of $J_{i,k}$ equals $\theta(v, w)$. This subsubgroup of $1 + |J_{i,k}|/\sqrt{\log m}$ processors does this in $O(\log m)$ time as follows:
   (a) If $|J_{i,k}| \geq \log m$, by giving to each of the $\sqrt{\log m}$ elements of $I_{i,k}$ (say, to element w) $|J_{i,k}|/\log m$ processors that w uses to find out, in $O(\log m)$ time, which element of $J_{i,k}$ equals $\theta(v, w)$.
   (b) If $|J_{i,k}| < \log m$, by partitioning $I_{i,k}$ into $1+|J_{i,k}|/\sqrt{\log m}$ equal pieces $I_{i,k,1}, I_{i,k,2}, \ldots$ (each of size roughly $\log m/|J_{i,k}|$) and giving each $I_{i,k,l}$ one processor. This processor sequentially finds $\theta(v, w)$ for all $w \in I_{i,k,l}$ in $O(\log m)$ time, since $|I_{i,k,l}||J_{i,k}| = O(\log m)$.

The fourth stage of the computation "fills in the blanks" by actually computing $\theta(v, w)$ for all $v \in L_A$ and $w \in L_B$. It does so with only $m^2/\log m$ processors by exploiting what was computed in the previous stages. Partition L_A into $m/\sqrt{\log m}$ contiguous blocks $X_1, X_2, \ldots$ of size $\sqrt{\log m}$ each. Similarly, partition L_B into $m/\sqrt{\log m}$ contiguous blocks $Y_1, Y_2, \ldots$ of size $\sqrt{\log m}$ each. Let Z_{ij} be the interval on the boundary common to A and B that is defined by the set of $\theta(v, w)$ such that $v \in X_i$ and $w \in Y_j$. Of course we already know the beginning and end of each such interval Z_{ij} (from the second and third stages of the computation). Furthermore, we have the following:

Lemma 6 $\sum_{i=1}^{m/\sqrt{\log m}} \sum_{j=1}^{m/\sqrt{\log m}} |Z_{ij}| = O(m^2/\sqrt{\log m})$.

Proof. See [2]. □

This lemma implies that we can afford to assign a group of $1 + |Z_{ij}|/\sqrt{\log m}$ processors to each pair X_i, Y_j. The task of this group is to compute $\theta(v, w)$ for all $v \in X_i$ and $w \in Y_j$ (of course each such $\theta(v, w)$ is in Z_{ij}). It suffices to show how such a group performs this computation in $O(\log m)$ time. If $|Z_{ij}| \leq \sqrt{\log m}$ then a single processor can solve the problem in $O((\sqrt{\log m})^2) = O(\log m)$ time, by the quadratic work method of Section 3. If $|Z_{ij}| > \sqrt{\log m}$ then we partition Z_{ij} into $|Z_{ij}|/\sqrt{\log m}$ pieces $J_1, J_2, \ldots$ of size $\sqrt{\log m}$ each. We assign to each J_k one processor which solves sequentially the sub-problem defined by X_i, J_k, Y_j, i.e. it computes for each $v \in X_i$ and $w \in Y_j$ the leftmost point of J_k through which passes a path that is shortest among the v-to-w paths that are constrained to go through J_k. This sequential computation takes $O(\log m)$ time (again, using the method of Section 3). It is done in parallel for all the J_k's. Now we must, for each pair v, w with $v \in X_i$ and $w \in Y_j$, select the best crossing point for it among the $|Z_{ij}|/\sqrt{\log m}$ possibilities returned by each of the above-mentioned sequential computations. This involves a total (i.e. for all such v, w pairs) of $O(|X_i||Y_j||Z_{ij}|/\sqrt{\log m}) = O(|Z_{ij}|\sqrt{\log m})$ comparisons, which can be done in $O(\log m)$ time by the $|Z_{ij}|/\sqrt{\log m}$ processors available (Brent's Theorem).

6. CRCW-PRAM algorithm

This subsection briefly sketches how the partitioning schemes of Subsection 5.2 translate into a CRCW-PRAM algorithm of time complexity $O(\log n (\log\log m)^2)$ and processor complexity $O(mn/\log\log m)$. Again, it suffices to show how $DIST_{A \cup B}$ can be obtained from $DIST_A$ and $DIST_B$ in $O((\log\log m)^2)$ time and with $m^2/\log\log m$ processors.

The procedure is recursive, and we describe it for the more general case when $DIST_A$ is $\ell \times h$ and $DIST_B$ is $h \times \ell$ (that is, $|L_A| = |L_B| = \ell$ and the common boundary has size h). It suffices to show that for any integer $q \leq h$ of our choice, $\ell h/q$ processors can, in $O((q + \log\log h)\log\log \ell)$ time, compute $\theta(v, w)$ for all $v \in L_A$ and $w \in L_B$. If we can show this then we are done because we can choose $q = \log\log h$, and we have $\ell = m$ and $h = m/2$.

The first stage of the computation partitions L_A into $\sqrt{\ell}$ contiguous blocks $X_1, X_2, \ldots$ of size $\sqrt{\ell}$ each. Similarly, L_B is partitioned into $\sqrt{\ell}$ contiguous blocks $Y_1, Y_2, \ldots$ of size $\sqrt{\ell}$ each. For each pair v, w such that v is an endpoint of an X_i and w is an endpoint of a Y_j, we assign h/q processors (we have enough processors because there are $O(\ell)$ such pairs). These processors compute, in $O(q + \log\log h)$ time, the point $\theta(v, w)$. Thus, if we let Z_{ij} denote the interval on the boundary common to A and B that is defined by the set $\theta(v, w)$ such that $v \in X_i$ and $w \in Y_j$, then after this stage of the computation we know the beginning and end of each such interval Z_{ij}.

The second stage of the computation "fills in the blanks" by doing, in parallel, ℓ recursive calls, one for each X_i, Y_j pair. The call for pair X_i, Y_j returns $\theta(v, w)$ for all $v \in X_i$ and $w \in Y_j$ (of course each such $\theta(v, w)$ is in Z_{ij}).

The time and processor complexities of the above method satisfy the recurrences: $T(\ell, h) \leq T(\sqrt{\ell}, h) + c_1(q + \log\log h)$, and $P(\ell, h) \leq \max\{c_2 \ell h/q, \sum_{i,j} P(\sqrt{\ell}, |Z_{ij}|)\}$, whose solutions are $T(\ell, h) = O((q + \log\log h)\log\log \ell)$ and $P(\ell, h) = O(\ell h/q)$ (using the fact that $\sum_{i,j}|Z_{ij}| = O(h\sqrt{\ell})$).

References

[1] A. Aggarwal and J. Park. Searching in Multidimensional Monotone Matrices. To be presented at FOCS 88.

[2] A. Apostolico, M.J. Atallah, L. Larmore and H.S. McFaddin. Efficient Parallel Algorithms for String Editing and Related Problems. Purdue TR CS-724 (Nov. 1987, revised July 1988).

[3] M.J. Atallah, R. Cole and M.T. Goodrich. Cascading Divide-and-Conquer: A Technique for Designing Parallel Algorithms, *Proc. 28th FOCS*, 1987, pp.151–160.

[4] R.P. Brent. The Parallel Evaluation of Arithmetic Expressions, *JACM*, 1987, pp.201–206.

[5] R. Cole. Parallel Merge Sort. *Proc. 27th FOCS*, 1986, pp.511–516.

[6] T.R. Mathies. A Fast Parallel Algorithm to Determine Edit Distance, CMU TR CS-130 (Apr. 1988).

[7] S. Ranka and S. Sahni. String Editing on an SIMD Hypercube Multicomputer, U. of Minnesota TR CS-29, (March 1988).

[8] R. A. Wagner and M. J. Fischer. The String to String Correction Problem, *JACM*, 1974, pp.168–173.

Fast Parallel Algorithms for Bipartite Permutation Graphs

Lin Chen*
Yaacov Yesha†

Department of Computer and Information Science
The Ohio State University
Columbus, Ohio 43210

Abstract

It is shown here that all the following problems:

- deciding whether a graph is a bipartite permutation graph, and if it is, obtaining its strong order;
- finding a Hamilton path/circuit in a bipartite permutation graph;
- solving a variant of the crossing number problem for a bipartite permutation graph;
- testing isomorphism of two bipartite permutation graphs;
- finding a maximum matching in a bipartite permutation graph

can be done in $O(\log^2 n)$ time with $O(n^3)$ processors. In addition, we show that a related problem, i.e., deciding whether a graph is an interval graph, and if it is, obtaining its interval representation can be done in $O(\log^2 n)$ time with $O(n^4)$ processors. Any of these findings either results in the first NC algorithm for a problem or greatly improves the previously known results.

1 Introduction

Bipartite graphs and permutation graphs are two important subclasses of perfect graphs [7]. Neither of these classes is contained in the other, and their intersection is nonempty. Spinrad, Brandstadt and Stewart [14] have shown that the graphs which are both bipartite and permutation graphs have good algorithmic properties. These graphs can be recognized efficiently in a sequential way. They have demonstrated that some problems which are NP-complete or of unknown complexity for graphs in either of the two larger classes can be solved in polynomial time when restricted to bipartite permutation graphs. In this paper, we further investigate the bipartite permutation graph. We give an NC algorithm for recognizing bipartite permutation graphs. The algorithm runs in $O(log^2 n)$

---

*Supported in part by the Office of Research and Graduate Studies of the Ohio State University.

†Supported in part by the National Science Foundation under grant DCR-8606366.

time with $O(n^3)$ number of processors on a concurrent read concurrent write parallel random access machine. There is an existing NC algorithm which recognizes permutation graphs [9]. From that algorithm, we can easily obtain an algorithm which recognizes bipartite permutation graphs, but this algorithm requires $O(log^3 n)$ time and $O(n^4)$ number of processors. So our algorithm runs much faster and needs much fewer processors. In addition, our algorithm is completely different from the old one. The previous algorithm decides permutation graphs by testing if a graph and its complement are both transitively orientable, while our algorithm recognizes bipartite permutation graphs by testing if there is a strong order. Our algorithm not only recognizes bipartite permutation graphs but also obtains a strong order if there is one. As we can see in [14], obtaining a strong order is the most important step in solving many problems for bipartite permutation graphs. We have successfully designed NC algorithms, for the first time, for several bipartite permutation graph problems. We are able to show that Hamilton path/circuit problem, a variant of the crossing number problem and graph isomorphism problem are all in NC if the input graphs are bipartite permutation graphs. Moreover, we can obtain a much more efficient NC algorithm for finding a maximum matching in a bipartite permutation graph. Furthermore, we give a much faster NC algorithm for recognizing an interval graph and constructing its interval representation.

2 Preliminaries

NC is a class of problems that can be solved in polylogarithmic time with polynomially many processors. For a graph $G = (V, E)$, we will often denote $|V|$ by n and $|E|$ by m. A graph is bipartite if its vertices can be partitioned into two sets such that each edge is between a vertex in one set and a vertex in the other. We will use the notation $G = (X, Y, E)$ to explicitly specify the two vertex sets. We will denote $|X|$ by n_1 and $|Y|$ by n_2. For ease of discussion, we assume the input graph is connected.

Let $G = (X, Y, E)$ be a bipartite graph. A (0, 1)-matrix B is called the bipartite graph matrix of G if B_{ij} is 1 if and only if $(x_i, y_j) \in E$. We say that a matrix M has the consecutive 1's property for rows if there exists a permutation matrix P such that the 1's in each row of MP occur consecutively. The consecutive 1's property for columns is defined in the similar way.

A graph is a permutation graph if there is some pair P, Q of permutations of the vertex set such that there is an edge between vertices x and y if and only if x precedes y in one of $\{P, Q\}$, while y precedes x in the other. A bipartite graph $G = (X, Y, E)$ is said to be strongly ordered if for all $(x, y), (x', y')$ in E, where x, x' are in X and y, y' are in Y, $x < x'$ and $y > y'$ imply (x, y') and (x', y) are in E. We call $X' \cup Y'$ a strong order of a bipartite graph $G = (X, Y, E)$, if X' and Y' are, respectively, permutations of X and Y, and (X', Y', E) is a strongly ordered bipartite graph.

The set of vertices adjacent to a vertex v is called the neighborhood of v, and is written as $N(v)$. X in a bipartite graph $G = (X, Y, E)$ has the adjacency property if for each vertex y in Y, $N(y)$ consists of vertices which are consecutive in X. X in a bipartite graph $G = (X, Y, E)$ has the enclosure property if for every pair of vertices v, w in Y such that $N(v)$ is a subset of $N(w)$, vertices in $N(w) - N(v)$ occur consecutively in X.

Vector a and vector b over {0, 1} are said to intersect if their inner product $a \cdot b > 0$. They overlap if they intersect and $a \cdot b < min(a \cdot a, b \cdot b)$. Vector a contains vector b if $a \cdot b = b \cdot b$.

An undirected graph $S(B) = (V, E)$ is called an intersection (or overlap) graph of a bipartite graph matrix B if there exists a bijection $f\colon V \longrightarrow$ the set of row vectors of B such that for any two distinct vertices v_i and v_j in V, $\{v_i, v_j\} \in E$ if and only if $f(v_i)$ and $f(v_j)$ intersect (or overlap).

A directed graph $T(B) = (V, E)$ is called the component graph of a bipartite graph matrix B, if there exists a bijection $f\colon V \longrightarrow$ the set of the components of the overlap graph such that for any two distinct vertices v_i and v_j in V, $(v_i, v_j) \in E$ if and only if a row vector x_p in the component corresponding to v_i contains a row vector x_q in the component corresponding to v_j.

A directed graph $C(B) = (V, E)$ is called a containment graph of a bipartite graph matrix B if there exists a bijection $f : V \longrightarrow$ the set of row vectors of B such that for any two distinct vertices v_i and v_j in V, $(v_i, v_j) \in E$ if and only if $f(v_i)$ contains $f(v_j)$. A containment (or component) graph is said to be reduced if we omit the arcs that are implied by transitivity.

The parallel computation model used in this paper is a parallel random access machine. Here we allow concurrent read and concurrent write. In case of concurrent write, we only allow several processors to write the same value into the same memory cell.

3 Some useful properties

In this section we present and investigate some properties of certain graphs and matrices which are helpful in the design of the efficient parallel algorithms.

Theorem 1 *If a bipartite graph matrix has the consecutive 1's property for rows, and if the overlap graph is connected, then there is a unique interval representation matrix for the corresponding intersection graph such that the size of the intersection of every two intervals (not necessarily distinct) equals that of the corresponding rows in the input matrix.*

Proof. see [1]. □

Lemma 1 *If M has the consecutive 1's property for rows, then PM also has the consecutive 1's property for rows, where P is any permutation matrix.*

Proof. There exists a permutation matrix P' such that the 1's in each row of MP' occur consecutively. So the 1's in each row of $(PM)P'(= P(MP'))$ also occur consecutively. Therefore, PM also has the property. □

In the following, we will use M_1, M_2 and M_3 to denote $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, $\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$ and $\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$, respectively.

Lemma 2 *$G = (X, Y, E)$ is a strongly ordered bipartite permutation graph if and only if the bipartite graph matrix B does not contain M_1, M_2 or M_3 as its submatrix.*

Proof. (Necessity part) Suppose G is a strongly ordered bipartite permutation graph. Consider any 2×2 submatrix $B' = \begin{bmatrix} b_{ip} & b_{iq} \\ b_{jp} & b_{jq} \end{bmatrix}$ of B. If $b_{iq} = 1$ and $b_{jp} = 1$, then $\{x_i, y_q\} \in E$ and $\{x_j, y_p\} \in E$. By the definition of strong order, $\{x_i, y_p\} \in E$ and $\{x_j, y_q\} \in E$. So $b_{ip} = 1$ and $b_{jq} = 1$. Hence B does not contain M_1, M_2 or M_3 as its submatrix.

(Sufficiency part) Assume no submatrix of B is M_1, M_2 or M_3. If $\{x_i, y_q\} \in E$ and $\{x_j, y_p\} \in E$, $i < j$ and $p < q$, then $b_{iq} = 1$ and $b_{jp} = 1$. It follows that $b_{ip} = 1$ and $b_{jq} = 1$. So $\{x_i, y_p\} \in E$ and $\{x_j, y_q\} \in E$. Therefore, the bipartite permutation graph is strongly ordered. □

Lemma 3 *If $G = (X, Y, E)$ is a strongly ordered bipartite permutation graph, then X has the adjacency property and enclosure property.*

Proof. See [14]. □

Corollary 1 *If $G = (X, Y, E)$ is a strongly ordered bipartite permutation graph, then Y has the adjacency property and enclosure property.*

Proof. By the above lemma, X has the adjacency and enclosure property if G is strongly ordered. Because of the symmetric property of a strong order, Y also has the adjacency and enclosure property. □

Theorem 2 *The following statements are equivalent for a bipartite graph $G = (X, Y, E)$:*

1. *G is a bipartite permutation graph.*
2. *There is a strong order of the graph.*
3. *There exists an ordering of X which has the adjacency and enclosure properties.*

Proof. See [14]. □

Corollary 2 *If $G = (X, Y, E)$ is a permutation graph, so is $G' = (X', Y', E')$, where $X' \subseteq X, Y' \subseteq Y$, and $E' = \{(x, y) : x \in X', y \in Y', (x, y) \in E\}$.*

Proof. Follows from the symmetry of a strong order. □

Theorem 3 *If a bipartite graph $G = (X, Y, E)$ is a permutation graph, then the bipartite graph matrix B has the consecutive 1's property for rows and columns.*

Proof. If (X, Y, E) is a permutation graph, then there is a strong order $X' \cup Y'$. Let B' be the bipartite graph matrix for (X', Y', E). Then there exist permutation matrices P_1 and P_2 such that $B' = P_1 B P_2$. Notice the 1's in each row and each column of B' must occur consecutively. Since $(P_1 B) P_2$ has consecutive 1's in each of its rows, $P_1 B$ has the consecutive 1's property for rows. Note that $B = {P_1}^{-1}(P_1 B)$. By Lemma 1, B also has the consecutive 1's property for rows. Likewise, B has the consecutive 1's property for columns. Hence the proof. □

If a bipartite permutation graph is strongly ordered, then there is at most one sequence of 1's in each row of the corresponding bipartite graph matrix. So each row can be represented by a pair of integers (a, b). (a, b) means that the sequence of 1's begins at the ath column and ends at the bth column. We define $(a_1, b_1) \leq (a_2, b_2)$ as $a_1 \leq b_1$ and $b_1 \leq b_2$.

Theorem 4 *A bipartite permutation graph $G = (X, Y, E)$ is strongly ordered if and only if the rows of the bipartite graph matrix B are linearly ordered by $\leq$, i.e., row $i \leq$ row j if $i < j$.*

Proof. (Necessity part) Suppose G is strongly ordered. Assume there exist i, j such that $i < j$ and row $i \not\leq$ row j. Let row i be represented by (p_1, q_1), row j by (p_2, q_2). Then either $p_1 > p_2$ or $q_1 > q_2$.

1. $p_1 > p_2$: Consider the submatrix $\begin{bmatrix} b_{ip_2} & b_{ip_1} \\ b_{jp_2} & b_{jp_1} \end{bmatrix}$. $b_{ip_2} = 0$ and $b_{ip_1} = 1$ since row i is (p_1, q_1). $b_{jp_2} = 1$ since row j is (p_2, q_2). Thus M_1 or M_3 is a submatrix of B, which contradicts the fact that G is strongly ordered.
2. $q_1 > q_2$: Consider the submatrix $\begin{bmatrix} b_{iq_2} & b_{iq_1} \\ b_{jq_2} & b_{jq_1} \end{bmatrix}$. A contradiction can be derived similarly.

Therefore, the rows of B must be in linear order.

(Sufficiency part) Suppose the rows of B are in linear order. Consider any 2×2 submatrix $\begin{bmatrix} b_{is} & b_{it} \\ b_{js} & b_{jt} \end{bmatrix}$, $i < j$ and $s < t$. Let row i be represented by (p_1, q_1), and row j by (p_2, q_2). If $b_{it} = 1$ and $b_{js} = 1$, then $q_1 \geq t$ and $p_2 \leq s$. Since row $i \leq$ row j, $p_1 \leq p_2 \leq s$. So $b_{is} = 1$. Likewise, $b_{jt} = 1$. Hence, B does not contain M_1, M_2 or M_3. It follows that G is strongly ordered. □

4 Algorithms

Recall that the input graph is assumed to be connected. So the reduced component graph is a tree. Let f be a bijection in the definition of the component graph.

Fulkerson and Gross [5] have designed an sequential algorithm which decides whether there exists a permutation Y' that has adjacency property, and obtains such a permutation if possible. Using a decomposition procedure in [8] as a subroutine, we have obtained an NC algorithm for this problem [1]. In this paper, we will give an algorithm which solves the problem of recognizing bipartite permutation graphs by way of deciding whether there exists a permutation Y' which has both adjacency and enclosure properties. In the following algorithm, we assume there are no identical rows in the bipartite graph matrix.

The algorithm works as follows. First consider the root r of the component graph. Check if there is a permutation of Y in the corresponding graph which has the adjacency property. If not, the input graph is not a permutation graph. Otherwise, check if the same permutation has the enclosure property. If not, then the input graph is not a permutation graph. If so, the graph is a permutation graph provided r does not have any children in the reduced component graph. If r has any children, then go on to the next step.

1. $f(r)$ contains only one vertex:
   (a) r has more than 2 children:
   The input graph is not a permutation graph.
   (b) r has 2 children in the reduced component graph:
   If for any vertex v other than r in the reduced component graph, v has at most one child and $f(v)$ contains only one vertex, then the input graph is a permutation graph. Otherwise, it is not.

(c) r has one child v in the reduced component graph:

   i. $f(v)$ contains only one vertex:

      If for any vertex v' in the reduced component graph, v' has at most one child and $f(v')$ contains only one vertex, then the input graph is a permutation graph. Otherwise, it is not.

   ii. $f(v)$ contains more than one vertex:

      If the submatrix corresponding to v is not a permutation graph, then the input graph is not a permutation graph. If is, check whether each interval begins at the leftmost position or ends at the rightmost position. If not, then the input graph is not a permutation graph, because no permutation can have the enclosure property considering the subgraph corresponding to r and v. Otherwise, go on to the next step.

      A. v has no children in the reduced component graph:

         The input graph is a permutation graph.

      B. v has one child v' in the reduced component graph:

         Check if the interval corresponding to v' is only contained in intervals either beginning at the leftmost position or ending at the rightmost position. If not, the input graph is not a permutation graph since no permutation has the enclosure property for the subgraph corresponding to r, v and v'. Otherwise, check for any vertex u other than r, v in the reduced component graph, if u has at most one child and $f(u)$ contains only one vertex. If so, the input graph is a permutation graph. If not, the input graph is not a permutation graph.

      C. v has two children v', v'' in the reduced component graph:

         Check if each of the intervals corresponding to v', v'' is only contained in intervals either beginning at the leftmost position or ending at the rightmost position. If not, the input graph is not a permutation graph since no permutation has the enclosure property for the subgraph corresponding to r, v, v' and v''. Otherwise, for every vertex u other than r, v in the reduced component graph, check if u has at most one child and $f(u)$ contains only one vertex. If so, the input graph is a permutation graph. Otherwise, the input graph is not a permutation graph.

2. $f(r)$ contains more than one vertex:

   First, notice that r has one or more children in the reduced component graph. If any of the intervals corresponding to this one or more children is not contained in the intervals either beginning at the leftmost position or ending at the rightmost position, then check if Y has the enclosure property for the submatrix corresponding to these intervals and r. If so, proceed. Otherwise, the input graph is not a permutation graph.

   Next, check if there exists any vertex v other than r in the reduced component graph such that v has more than one child or $f(v)$ contains more than one vertex. If so, the input graph is not a permutation graph. Otherwise, the input graph is a permutation graph.

The above algorithm recognizes bipartite permutation graphs. It can be easily extended to compute the strong order as well. What we need to do is to put every interval either at the leftmost possible position or rightmost possible position. It follows from theorem 1 and theorem 4, a strong order can be obtained by sorting the intervals. In [1], Chen and Yesha have shown, for the first time, that testing for the consecutive 1's property is in NC. In fact, Chen and Yesha [1] have shown the following thing.

Theorem 5 *Testing if a* $(0,1)$*-matrix has the consecutive 1's property for rows, and if so, transforming the matrix into one with consecutive 1's in each of its rows can be done in* $O(\log n + \log^2 m)$ *time with* $O(m^2n + m^3)$ *number of processors, where* $m \times n$ *is the size of the matrix.*

Therefore, in the above algorithm, checking if the matrix has the consecutive 1's property for rows can be done in $O(log^2 n)$ time with $O(n^3)$ processors. For each subgraph corresponding to a vertex of the component graph, checking if there is a permutation which has the enclosure property can be done in constant time with $O(n)$ processors. We can sort the intervals in $O(logn)$ time with $O(n)$ processors [2]. All the other checking can be done in a straightforward way. Thus we have the following theorem:

Theorem 6 *Deciding whether a graph is a bipartite permutation graph, and if it is, obtaining its strong order can be done in* $O(log^2 n)$ *time with* $O(n^3)$ *processors.*

In [14], there is a sequential algorithm which finds a Hamilton path in a bipartite permutation graph if the path exists.

Instance: a bipartite permutation graph G.
Question: does G have a Hamilton path?

```
1. obtain a strong order $X \cup Y$ of $G$;
2. if $x_1, y_1, ..., x_{k-1}, y_{k-1}, x_k, (y_k)$ is a path, then return('yes');
3. if $y_1, x_1, ..., y_{k-1}, x_{k-1}, y_k, (x_k)$ is a path, then return('yes');
4. return('no')
```

The dominating step is to obtain a strong order. Now that we have designed an NC algorithm for obtaining a strong order, the problem of finding a Hamilton path in a bipartite permutation graph can be solved in $O(log^2 n)$ time with $O(n^3)$ processors.

Next we describe the algorithm for Hamilton circuit problem.

Instance: a bipartite permutation graph G.
Question: does G have a Hamilton circuit?

```
1. obtain a strong order $X \cup Y$ of $G$;
2. if $x_i, y_i, x_{i+1}, y_{i+1}$ is a cycle of length four for $1 \le i < k$ then return('yes')
                                                                        else return('no')
```

Obviously, the resource requirements are of the same order as the above. Moreover, we can easily modify the above two algorithms to actually obtain a Hamilton path/circuit, if one exists, within the same resource bound.

Now let us take a look at a variant of the crossing number problem as described in [6] and [10]. Given a bipartite graph $G = (X, Y, E)$ and a number k, can G be embedded in a unit square so that all vertices from X are on the north boundary, all vertices from Y are on the south boundary, and the number of crossings is k or less?

In [14], Spinrad, Brandstadt and Stewart have shown the above problem can be solved for bipartite permutation graphs by constructing the strong order and counting the number of crossings. We now give an efficient implementation of their idea. Since we have already solved the problem of constructing a strong order, we only need consider the way to count the number of crossings.

Let M be a bipartite graph matrix for a strongly ordered bipartite permutation graph $G = (X, Y, E)$. Then $(x_i, y_j) \in E$ if and only if $m_{ij} = 1$. We will use $M(i : j, p : q)$ to denote a submatrix of M which consists of row i to row j and column p to column q of M. Suppose $m_{ij} = 1, m_{pq} = 1$ and $i < p$. The two corresponding edges cross if and only if $j > q$. It can be readily seen that, if $m_{ij} = 1$, then number of edges crossing the edge corresponding to m_{ij} is the number of 1's in $M(1 : i-1, j+1 : n)$ and $M(i+1 : m, 1 : j-1)$. We denote this number by $c(m_{ij})$. In addition, we denote by $n(M)$ the number of 1's in the matrix M. Notice every crossing is associated with two edges. So the total number of crossings is $\sum_{i,j} c(m_{ij})/2$. To make the computation simpler, we define another quantity $r(m_{ij})$ as follows:

$$r(m_{ij}) = \begin{cases} 0 & \text{if } m_{ij} = 1 \\ \text{n(M(i+1:m, 1:j-1))} & \text{otherwise} \end{cases}$$

Now the total number of crossings is $\sum_{i,j} r(m_{ij})$.

Obviously , we have

$$n(M(i+1 : m, 1 : j-1)) = \sum_{k=i+1}^{m} n(M(k : k, 1 : j-1))$$

Consider $n(M(k : k, 1 : j-1))$. If the 1 in the kth row of M begins at the sth element, then

$$n(M(k : k, 1 : j-1)) = \begin{cases} 0 & \text{if } s \geq j \\ j-s & \text{otherwise} \end{cases}$$

The places where 1 begins for each of the rows can be found in $O(n^2)$ sequential time. Then the total number of crossings can be computed in $O(n^2)$ sequential time.

Next let us take a look at the parallel algorithm for this problem. We can use Shiloach and Vishkin's technique [13] to locate the first column where 1 begins for each row. This can be done in constant time with $O(n^2)$ processors. The total number of crossings can be computed in $O(logn)$ time with $O(n^2)$ processors. So the resource bounds are determined by the procedure for obtaining the strong order.

Testing isomorphism of two bipartite permutation graphs can be done by comparing the strong orders. The detailed description can be acquired without much difficulty. It is already known that, since bipartite permutation graphs are a subclass of co-comparability graphs, we can use the algorithm for finding a maximum matching of a co-comparability graph to solve the maximum matching problem for bipartite permutation graphs [9]. This algorithm requires at least $O(log^3 n)$ time and n^5 number of processors. Here we have found that there is a much more efficient NC algorithm which solves the maximum matching problem for bipartite permutation graphs. Notice that bipartite permutation graphs are also a subclass of transformable convex bipartite graphs.

Transformable convex bipartite graphs [1] are the closure of convex bipartite graphs [4] with respect to graph isomorphism. Therefore, to solve the problem for bipartite permutation graphs, we can simply use the algorithm for transformable convex bipartite graphs [1], which only takes $O(log^2 n)$ time and $O(n^3)$ number of processors.

Finally, we will show that, for a related problem, interval graph recognition problem, we can design a much faster NC algorithm. Kozen, Vazirani and Vazirani [11] gave the first NC algorithm for this problem. Another NC algorithm can be obtained by combining the results of Naor, Naor and Schaeffer [12] and Helmbold and Mayr [9]. We can recognize an interval graph by testing if a graph is a chordal graph and if the complement of the graph is a comparability graph. The second step is the dominating step. It takes $O(\log^3 n)$ time and $O(n^4)$ processors. Kozen, Vazirani and Vazirani did not give the analysis of their algorithm, but it can be readily seen that their algorithm takes at least $O(\log^3 n)$ time and $O(n^4)$ processors, since their algorithm uses as a subroutine the procedure which decides if a graph is a comparability graph.

Our new algorithm is quite different from the above two algorithms, and can be outlined in the following way:

1. Test if G is a chordal graph. If not, then G is not an interval graph, and stop.
2. Compute all the maximal cliques of G.
3. Check if the vertex versus maximal clique incidence matrix has the consecutive 1's property for rows. If so, then G is an interval graph. Otherwise, it is not.

Testing the chordality can be completed in $O(\log^2 n)$ time with $O(mn^2)$ processors[12]. The second step can be done in no more than $(\log^2 n)$ with $O(n^4)$ processors[3][12]. By Theorem 5, the last step can be done in $O(\log^2 n)$ time with $O(n^3)$ processors. We can actually obtain the interval representation in this step, if one exists. Therefore, the total resource requirements are $O(\log^2 n)$ time and $O(n^4)$ processors.

Acknowledgment

We would like to thank Alejandro Schaffer of Stanford University for bringing to our attention the existing NC algorithm for recognizing permutation graphs.

References

[1] L. Chen, Y. Yesha, "Efficient parallel algorithms for recognizing transformable convex bipartite graphs and finding their maximum matchings", OSU-CISRC-8/88-TR 26, The Ohio State University, August 1988.

[2] R. Cole, "Parallel merge sort", Proc. 27th IEEE Symposium on Foundations of Computer Science, 511-517, 1986.

[3] E. Dahlhaus, M. Karpinski, "Fast parallel computation of perfect and strongly perfect elimination schemes", IBM Research Report RJ5901, October 1987.

[4] E. Dekel, S. Sahni, "A parallel matching algorithm for convex bipartite graphs", Proc. International Conference on Parallel Processing, 178-184, 1982.

[5] D.R. Fulkerson, O.A. Gross, "Incidence matrices and interval graphs", Pacific J. Math., Vol. 15, 835-855, 1965.

[6] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman & Co., San Francisco, 1979.

[7] M.C. Golumbic, Algorithmic Graph Theory and Perfect Graphs, Academic Press, 1980.

[8] X. He, Y. Yesha, "Parallel recognition and decomposition of two terminal series parallel graphs", Information and Computation, Vol. 75, 15-38, Oct. 1987.

[9] D. Helmbold, E. Mayr, "Perfect graphs and parallel algorithms", Proc. International Conference on Parallel Processing, 853- 860, 1986.

[10] D.S. Johnson, "The NP-completeness column: an ongoing guide", J. Algorithm, Vol. 3, 89-99, 1982.

[11] D. Kozen, U.V. Vazirani, V.V. Vazirani, "NC algorithms for comparability graphs, interval graphs, and unique perfect matchings", Fifth Annual Foundations of Software Technology and Theoretical Computer Science Conference, Lecture Notes in Computer Science, Vol. 206, 496-503, Springer-Verlag, 1985, and (in revised form) TR86-799, Dept. of Computer Science, Cornell University, December 1986.

[12] J. Naor, M. Naor, A.A. Schaeffer, "Fast parallel algorithms for chordal graphs", Proc. 19th Ann. ACM Symp. on Theory of Computing, 355-364, 1987.

[13] Y. Shiloach, U. Vishkin, "Finding the maximum, merging, and sorting in a parallel computation model", J. Algorithms, Vol. 2, 88-102, 1981.

[14] J. Spinrad, A. Brandstadt, L. Stewart, "Bipartite permutation graphs", Discrete Applied Math., Vol. 18, 279-292, 1987.

PT-OPTIMAL ALGORITHMS FOR INTERVAL GRAPHS

ABHA MOITRA[1,2] and RICHARD JOHNSON[2]
Department of Computer Science, Cornell University, Ithaca, New York 14853

Abstract: In this paper, we consider parallel algorithms on *interval graphs.* An interval graph is a graph having a one-to-one correspondence with a sequence of intervals on the real line, such that each vertex maps to an interval in the sequence and an edge exists between two vertices if and only if the corresponding intervals overlap. Throughout the paper we use the CREW PRAM model and assume interval representations of graphs are given. We give PT-optimal algorithms for maximum weighted clique, maximum independent set, minimum clique cover, and minimum dominating set for interval graphs; and Hamiltonian circuit for proper interval graphs. In addition, we present algorithms for depth-first search of interval graphs and maximum matching on proper interval graphs, and we give an improved algorithm for minimum bandwidth on proper interval graphs.

1 INTRODUCTION

In this paper, we consider parallel algorithms on *interval graphs.* An interval graph is a graph having a one-to-one correspondence with a sequence of intervals on the real line, such that each vertex maps to an interval in the sequence and an edge exists between two vertices if and only if the corresponding intervals overlap. We denote the interval set as $I = \{i_1, i_2, \ldots, i_n\}$ where i_k denotes the real line interval $[a_k, b_k]$. An interval graph is *proper* if no interval is properly contained within some other interval. In this paper, we will refer to interval i, interval $[a_i, b_i]$, and vertex i (corresponding to interval i) interchangeably. Given a graph, its interval representation (if one exists) may be obtained on the CRCW PRAM model with the $O(\log^2 n)$ time and $O(n+m)$ processor algorithm of Klein [8] or with the $O(\log n)$ time and $O(n^3)$ processor algorithm of Novick [11]. There are many applications of interval graphs, spanning the study of problems in genetics to archeology; for more details see Golumbic [6] where he says "interval graphs are among the most useful mathematical structures for modeling real world problems." In many applications, a representation of an interval graph is part of the problem statement, and hence it is natural to study the complexity of parallel algorithms on representations of interval graphs.

We improve several parallel algorithms from Bertossi and Bonuccelli [2] by giving PT-optimal algorithms for maximum weighted clique, maximum independent set, minimum clique cover, and minimum dominating set for interval graphs; and Hamiltonian circuit for proper interval graphs. We also present $O(\log n)$ time, $O(n^2/\log n)$ processor algorithms for depth-first search of interval graphs and maximum matching on proper interval graphs, and we give

---

[1] Current address: Odyssey Research Associates, Ithaca, New York 14850

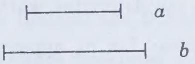
[2] Research supported in part by National Science Foundation grant DCR-86-02307.

an improved algorithm for minimum bandwidth for proper interval graphs.

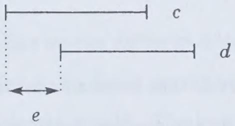
Bertossi and Bonuccelli [2] give several parallel algorithms for interval graphs that rely on parallel computation of $right(i)$ and $next(i)$ for each interval i. Assume I is sorted by increasing right endpoints; then $right(i)$ is the rightmost ending interval overlapping interval i, that is, $right(i) = j$ such that $b_j = \max\{b_k \mid i < k, a_k < b_i \le b_k\}$, $= 0$ otherwise (this definition is slightly different from that in [2]). Note that the definition requires I sorted so as to break ties consistently. $next(i)$ is the leftmost ending interval beginning after the end of interval i, that is, $next(i) = j$ such that $b_j = \min\{b_k \mid b_i < a_k\}$, $= 0$ otherwise. Bertossi and Bonuccelli give algorithms for $right(i)$ and $next(i)$ (for all i) which take $O(\log n)$ time and $O(n^2/\log n)$ processors. In Section 2, we show that $right(i)$ and $next(i)$ for all i can be computed in $O(\log n)$ time with $O(n)$ processors. In Section 3 we use $right(i)$ and $next(i)$ to develop several PT-optimal algorithms for representations of interval graphs.

2 FASTER COMPUTATION OF *right*(*i*) AND *next*(*i*)

To motivate our algorithm for computing $right(i)$, consider an interval a properly contained in some interval b.



Clearly, no interval i will choose $right(i) = a$ since if i overlaps a then i also overlaps b, and b ends right of a. After discarding intervals properly contained in another interval, we are left with a proper interval graph. In the example below, any interval ending in e will not choose d but may choose c.



To compute $right(i)$ for all i, perform the following steps:

Step 1 Form set P from I by discarding properly contained intervals in I, i.e., P is a smallest sized subset of I such that for all $i \in I$, there exists $j \in P$ such that j contains i.

Step 2 Sort the left endpoints of intervals in P to obtain $P' = \langle a_{\sigma(1)}, a_{\sigma(2)}, \ldots, a_{\sigma(h)}\rangle$.

The crucial property of intervals in P is that for any $i, j \in P, i \neq j$, if interval i starts before interval j, then interval i ends before interval j. Therefore, for $i \in I$, if $\mathit{right}(i) \neq 0$ then $\mathit{right}(i) = \sigma(k)$ where i ends in the interval $[a_{\sigma(k)}, a_{\sigma(k+1)})$. Since the intervals $[a_{\sigma(k)}, a_{\sigma(k+1)})$ are sorted and nonoverlapping, the search can be done in $O(\log n)$ time for each interval. The total cost of this step is $O(\log n)$ time, $O(n)$ processors.

We must now consider how to implement Step 1 efficiently in parallel.

1. Let $E = \{(a,l) \mid \exists k \in I : a_k = a\} \cup \{(b,r) \mid k \in I : b_k = b\}$, the set of left and right endpoints of intervals in I. Sort the elements of E on the first component, forming list L_{lr}. Form list L_l from list I by sorting elements of I by increasing left endpoints, and then sorting by decreasing right endpoints all subsequences of elements having the same left endpoint. Note that elements in L_l may have the same left and/or right endpoints; for this reason, we maintain a mapping onto I so that the corresponding vertices are known. Both lists may be formed in $O(\log n)$ time using $O(n)$ processors.

2. Construct a list $L_{l'}$ from L_l by replacing each element in L_l by the index of its right endpoint in L_{lr}. This can be done by binary search in $O(\log n)$ time using $O(n)$ processors.

3. Now, for any interval $k \in I$, we can determine whether it is properly contained within some other interval of I as follows: Let a_k appear as the j^{th} element in L_l and let the j^{th} element of $L_{l'}$ be p. k is properly contained within some other interval of I iff $\exists\, j', j' < j$ and $L_{l'}[j'] \geq p$. (If such a j' exists, then there exists an interval that starts no later than k since $j' < j$, and that interval ends not before interval k since $L_{l'}[j'] \geq p$.) So, perform a parallel prefix computation to obtain the largest value in every prefix of $L_{l'}$. Now for each $2 \leq i \leq n$, the interval corresponding to $L_{l'}[i]$ is to be discarded iff $\max\{L_{l'}[1 \ldots i-1]\} \geq L_{l'}[i]$. This can be done in $O(\log n)$ time using $O(n/\log n)$ processors.

Similarly, we can identify all intervals that cover another interval. (Here, we would perform a parallel prefix computation to obtain the smallest value in every suffix of $L_{l'}$.)

As with $\mathit{right}(i)$, $\mathit{next}(i)$ may be computed more efficiently if we first discard some intervals. If interval b properly contains a, then a will always be chosen over b for $\mathit{next}(i)$ since a ends left of b. After discarding all intervals that cover another interval, we again are left with a proper interval graph. To compute $\mathit{next}(i)$, perform the following steps:

Step 1 Form set P from I by discarding intervals that properly contain other intervals from I; i.e., P is the smallest sized subset of I such that for all $i \in I - P$ there exists $j \in P$ such that i contains j.

Step 2 Sort the left endpoints of intervals in P, and denote this sorted list by $P' = \langle a_{\sigma(1)}, a_{\sigma(2)}, \ldots, a_{\sigma(h)} \rangle$. The crucial property of intervals in P is that for any $i, j \in P, i \neq j$, if interval i starts before interval j, then interval i ends before interval j. Therefore, for $i \in I$, if *next*$(i) \neq 0$ then *next*$(i) = \sigma(k+1)$ where i ends in the interval $[a_{\sigma(k)}, a_{\sigma(k+1)})$. Since the intervals $[a_{\sigma(k)}, a_{\sigma(k+1)})$ are sorted and nonoverlapping, the search can be done in $O(\log n)$ time for each interval. The total cost of Step 2 is $O(\log n)$ time, $O(n)$ processors.

We have seen that Step 1 can be computed in $O(\log n)$ time using $O(n)$ processors, so both *right*(i) and *next*(i) for all i can be computed in $O(\log n)$ time using $O(n)$ processors.

3 SEVERAL PT-OPTIMAL ALGORITHMS FOR INTERVAL GRAPHS

In this section we present parallel algorithms for maximum weighted clique, maximum independent set, minimum clique cover, and minimum cardinality dominating set for interval graphs; and Hamiltonian circuit for proper interval graphs. Each of these is an improvement over the corresponding algorithm in Bertossi and Bonuccelli [2] and is now PT-optimal.

Given an interval graph with weights associated with the vertices, we now describe how to find a *maximum weighted clique.* First of all we can restrict ourselves to the case where all weights are positive. This is because if all the weights are non-positive then the maximum weighted clique consists simply of a single vertex with maximum weight. Otherwise, no non-positive vertex weight can be part of a maximum weighted clique and hence all the vertices with non-positive weights may be dropped. This preprocessing may be done in $O(\log n)$ time using $O(n/\log n)$ processors.

With each point p on the real line we can define its weight to be the sum of the weights of all the intervals that contain the point p. To find a maximum weighted clique it is enough to find a point on the real line with maximum weight. The weights of points on the real line change only when either an interval starts or an interval ends and hence maximum weighted clique can be found as follows. Sort the $2n$ endpoints $a_1, \ldots a_n, b_1, \ldots, b_n$ in increasing order and assign positive weights to left endpoints and negative weights to right endpoints. The clique weights can be found by computing the partial sums and then the maximum weighted clique can be found. The sorting can be done in $O(\log n)$ time using $O(n)$ processors [5]; the partial sums can be obtained in $O(\log n)$ time using $O(n)$ processors [9]. The maximum weighted clique can, therefore, be found in $O(\log n)$ time using $O(n)$ processors.

An *independent set* of a graph is a subset of its vertices such that no two vertices in the subset are connected by an edge. Given an interval set I, we define *first*$(I) = j$, such that $b_j = \min\{b_k\}$. For interval graphs a maximum cardinality independent set S can be defined inductively as: *first*$(I) \in S$; if $i \in S$, then *next*$(i) \in S$. So if *next*(i) has been computed for all $i \in I$ then S can be obtained by the standard parallel doubling technique. The entire computation requires $O(\log n)$ time using $O(n)$ processors.

In the *minimum clique cover* problem, a minimum cardinality partitioning of the nodes of the graph into cliques is required. For interval graphs this can be done by augmenting its maximum independent set. If $S = \{i_1, \ldots, i_m\}$ is a maximum independent set, then there exists a minimum clique cover $C_{i_1}, \ldots, C_{i_m}$ such that interval $i_j \in S$ belongs to C_{i_j} and each $h \notin S$ belongs to C_{i_k} iff

$$b_{i_k} = \min\{b_i \mid i \in S \text{ and } a_h < b_i \leq b_h\}$$

For each vertex h, the clique to which it belongs can be found in $O(\log n)$ time by doing binary search on S sorted by right endpoints. The minimum clique cover can therefore be found in $O(\log n)$ time using $O(n)$ processors.

A *dominating set* for a graph is a subset D of its vertices such that every vertex of the graph is either in D or is connected by an edge to a vertex in D. The sequential greedy algorithm of Booth and Johnson [3] finds a minimum cardinality dominating set for interval graphs as follows: *right*(*first*$(I)) \in D$; if $i \in D$, then *right* (*next*$(i)) \in D$ where *right*$(0) = 0$. The standard doubling technique leads to an $O(\log n)$ time, $O(n)$ processor algorithm for minimum cardinality dominating set.

The algorithms given here for maximum weighted clique, maximum independent set, minimum clique cover and minimum cardinality dominating set can be shown to be PT-optimal as follows. A sequential lower time bound of $\Omega(n \log n)$ can be shown for the problem of determining whether n real line intervals are pairwise disjoint [13]. Gupta, Lee and Leung [7] have shown a sequential lower time bound of $\Omega(n \log n)$ for maximum weighted clique, maximum independent set, and minimum clique cover on interval graphs (with representation given) by arguing that a solution to any one of these problems also solves the problem of determining whether n real line intervals are pairwise disjoint. A similar argument applies to minimum cardinality dominating set.

A graph has a *Hamiltonian circuit* if there is a circuit in the graph where each vertex of the graph appears exactly once. In Bertossi [1] it is shown that a Hamiltonian circuit exists for an interval graph if and only if intervals in $I - \{$*first*$(I)\}$ can be partitioned into disjoint subsets J and K such that:

(1) *right*(*first*(I)) $\in J$, and if $j \in J$, then *right*(j) $\in J$, and this process should result in *last*(I) $\in J$.

(2) the subgraph represented by $K \cup \{$*first*(I), *last*(I)$\}$ has a Hamiltonian path.

Further, Bertossi [1] has shown that for proper interval graphs, a Hamiltonian path exists if and only if consecutive intervals overlap when the intervals are sorted on left endpoints. The partitioning of the intervals can be done in $O(\log n)$ time using $O(n)$ processors. The two conditions on this partitioning can also be checked within these bounds. If the partitioning can be done, then the Hamiltonian circuit is formed by putting together the Hamiltonian path on $K \cup \{$*first*(I), *last*(I)$\}$ with the path formed by sorting elements of J on right endpoints. The entire computation requires $O(\log n)$ time using $O(n)$ processors, which is PT-optimal, as Weide [14] has shown an $\Omega(n \log n)$ lower bound for sequential algorithms for Hamiltonian circuit on proper interval graphs.

4 OTHER PARALLEL ALGORITHMS ON INTERVAL GRAPHS

In [10] we gave an $O(\log^2 n)$ time, $O(n^6/\log n)$ processor algorithm for maximum matching on interval graphs; we now show that when working with representations of proper interval graphs, a maximum matching can be found much more efficiently. We first give an algorithm for depth-first search (DFS) on representations of interval graphs and then show how the DFS algorithm can be used to find a maximum matching on representations of proper interval graphs. Additionally, we give an improved algorithm for minimum bandwidth on proper interval graphs.

Depth-First Search

In this section we show that a DFS tree of an interval graph representation can be obtained in $O(\log n)$ time using $O(n^2/\log n)$ processors.

Given an interval set I, let the intervals be sorted by right endpoints so that $b_i \leq b_{i+1}$ for $1 \leq i < n$. We define *last*(I) $= j$, such that $b_j = \max\{b_k\}$. For each interval $i \in I$, we define *parent*(i) $= j$, if $b_j = \min\{b_k \mid k > i, a_k \leq b_i \leq b_k\}, = 0$, otherwise. *parent*($i$) is the interval overlapping i which ends first after the end of i, if it exists. Given a connected interval graph G, construct $T = (V, E')$ where $E' = \{(i, \mathit{parent}(i)) \mid \mathit{parent}(i) \neq 0\}$.

Theorem 1: For an interval graph G, T (as described above) is a DFS tree and can be constructed in $O(\log n)$ time using $O(n^2/\log n)$ processors on the CREW PRAM model.

proof: We show that T is a spanning tree with no cross edges. Note that the root of T is *last*(I). For each vertex i (other than the root), *parent*(i) is the parent of i in T.

1. Suppose there exists interval $i, i \neq n$, such that *parent*$(i) = 0$. Then i does not intersect any intervals in the set $S = \{i+1, i+2, \ldots, n\}$. Additionally, no interval ending before b_i can intersect an interval in S. So there can be no edge between the sets S and $I - S$. This contradicts G connected. Hence, *parent*(i) exists for every interval other than n; i.e., every vertex other than the root has a parent. Since each edge in T joins an interval to one ending later, there can be no cycles. Since every vertex other than the root contributes an edge, there are $n-1$ edges. Thus T is a spanning tree.

2. Suppose some edge $e = (i,j)$ in $E - E'$ is a cross edge. Then $[a_i, b_i]$ overlaps $[a_j, b_j]$. Without loss of generality, assume $b_i \leq b_j$. Let k be the least common ancestor of i and j in T. Then $b_i \leq b_k$ and $b_j \leq b_k$. Let $i, i_1, i_2, \ldots, i_m, k$ be the vertices on the path between i and k. Then $b_i \leq b_{i_1} \leq b_{i_2} \leq \ldots \leq b_{i_m} \leq b_k$. Since *parent*$(i) = i_1, b_{i_1} \leq b_j$. Since (i,j) is an edge, intervals i and j overlap; so i_1 and j overlap. Now (i_1, j) is a cross edge. Repeating this argument for each vertex on the path from i to k results in $b_k \leq b_j$, a contradiction. Thus (i,j) cannot be a cross edge.

The time to compute the above DFS tree for a connected graph G is simply the time to compute *parent*(i) for all i in I. For any i, *parent*(i) can be calculated in $O(\log n)$ time using $O(n/\log n)$ processors by simply finding the minimum right endpoint of all intervals that overlap with i and end after i. The total cost of computing the DFS is then $O(\log n)$ time using $O(n^2/\log n)$ processors on the CREW PRAM model. □

Maximum Matching on Proper Interval Graphs

In the case of proper interval graphs, maximum matching is easily solved via depth-first search.

Theorem 2: In a proper interval graph G, a maximum matching can be found in $O(\log n)$ time using $O(n^2/\log n)$ processors on the CREW PRAM model.

proof: Assume G is connected. Let set I be sorted by right endpoints. Since G is proper, each interval i except *last*(I) intersects $i+1$. Hence the edges of the DFS tree are $E = \{(i, i+1) \mid 1 \leq i < n\}$. Clearly, these edges form a path of length $n-1$ and there are $\lfloor \frac{n}{2} \rfloor$ odd edges on this path. The odd edges form a matching, and this matching is maximum since no matching on any graph of n vertices can exceed size $\lfloor \frac{n}{2} \rfloor$. Now consider a proper interval graph G having more than one connected component. Applying DFS results in a unary spanning tree of each component. By a simple doubling strategy, we can locate the odd edges in all of these spanning trees; these edges form maximum matchings on the connected components. Since the components are independent, these edges form a maximum matching on the entire graph.

Thus, a maximum matching on a representation of a proper interval graph can be obtained in $O(\log n)$ time using $O(n^2/\log n)$ processors. □

Minimum bandwidth

A graph $G = (V, E)$ has a *bandwidth* h if there is a function $f : V \to 1, \ldots, |V|$ such that for all $(u, v) \in E$ $|f(u) - f(v)| \leq h$. The *minimum bandwidth* problem is to find the smallest such h. For $1 \leq i \leq n$, let an *overlap clique* C_i of I be a maximal set of intervals all overlapping at a_i. For proper interval graphs, Orlin, Bonuccelli, and Bovet [12] have shown that if the intervals are indexed by increasing right endpoint then the intervals within each overlap clique C_i are consecutive, that is, $C_i = \{j, j+1, \ldots, k\}$ for some j and k. From this it follows that the minimum bandwidth is one less than the size of the largest overlap clique. Since a maximum clique is an overlap clique, the bandwidth for proper interval graphs can be obtained as a byproduct of finding the maximum clique in $O(\log n)$ time using $O(n)$ processors.

Recently, Klein [8] has given $O(\log^2 n)$ time, $O(n + m)$ processor algorithms on the CRCW PRAM model for the following interval graph problems: recognition, maximum weighted clique, maximum independent set, optimal coloring, breadth-first search, depth-first search, and graph isomorphism. Note that these algorithms assume an adjacency list representation and thus have different sequential lower bounds than algorithms assuming an interval representation.

The following tables summarize our results for several parallel algorithms for interval graphs.

PT-optimal Parallel Algorithms for Interval Graph Representation

	Bertossi and Bonuccelli [2]		*present work*	
Algorithm	*time*	*processors*	*time*	*processors*
maximum weighted clique	$O(\log n)$	$O(n^2/\log n)$	$O(\log n)$	$O(n)$
maximum independent set	$O(\log n)$	$O(n^2/\log n)$	$O(\log n)$	$O(n)$
minimum clique cover	$O(\log n)$	$O(n^2/\log n)$	$O(\log n)$	$O(n)$
minimum dominating set	$O(\log n)$	$O(n^2/\log n)$	$O(\log n)$	$O(n)$
Hamiltonian circuit of proper interval graphs	$O(\log n)$	$O(n^2/\log n)$	$O(\log n)$	$O(n)$

Other Parallel Algorithms for Interval Graph Representations

Algorithm	*time*	*processors*
depth-first search	$O(\log n)$	$O(n^2/\log n)$
maximum matching on proper interval graphs	$O(\log n)$	$O(n^2/\log n)$
maximum matching	$O(\log^2 n)$	$O(n^6/\log n)$
minimum bandwidth of proper interval graphs	$O(\log n)$	$O(n)$

References

[1] A. A. Bertossi. Finding hamiltonian circuits in proper interval graphs. *Information Processing Letters 17*, pages 97–101, 1983.

[2] A. A. Bertossi and M. A. Bonuccelli. Some parallel algorithms on interval graphs. *Discrete Applied Mathematics 16*, pages 101–111, 1987.

[3] K. Booth and J. H. Johnson. Dominating sets in chordal graphs. *SIAM Journal of Computing 11*, pages 191–199, 1982.

[4] E. G. Coffman and R. L. Graham. Optimal scheduling for two-processor systems. *Acta Informatica 1*, pages 200–213, 1972.

[5] R. Cole. Parallel merge sort. In *27th FOCS*, pages 511–516, 1986.

[6] M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs.* Computer Science and Applied Mathematics. Academic Press, Inc., 1980.

[7] U. I. Gupta, D. T. Lee, and Y. T. Leung. Efficient algorithms for interval graphs and circular-arc graphs. *Networks 12*, pages 459–467, 1982.

[8] P. Klein. Efficient parallel algorithms for chordal graphs. Laboratory for Computer Science, MIT, 1988.

[9] C. P. Kruskal, L. Rudolph, and M. Snir. The power of parallel prefix. *IEEE Transactions on Computers, C-34*, pages 965–968, 1985.

[10] A. Moitra and R. C. Johnson. Parallel algorithms for maximum mathing and other problems on interval graphs. Cornell University technical report TR 88-927, 1988.

[11] M. Novick. Personal communications.

[12] J. B. Orlin, M. A. Bonuccelli, and D. P. Bovet. An $O(n^2)$ algorithm for coloring proper circular arc graphs. *SIAM Journal of Algebraic Discrete Methods 2*, pages 88–93, 1981.

[13] F. P. Preparata and M. I. Shamos. Computational geometry: An introduction. In *Texts and Monographs in Computer Science.* Springer-Verlag, 1985.

[14] B. Weide. A survey of analysis techniques for discrete algorithms. *Computing Surveys 9*, pages 291–313, 1977.

FAST PARALLEL ALGORITHMS FOR TREE DECOMPOSING AND PARSING PARTIAL K-TREES

N. CHANDRASEKHARAN
DEPARTMENT OF MATHEMATICAL SCIENCES
Clemson University
Clemson, South Carolina 29634.

S.T. HEDETNIEMI
DEPARTMENT OF COMPUTER SCIENCE
Clemson University
Clemson, South Carolina 29634.

Abstract

In this paper we present parallel algorithms for constructing a tree decomposition (as introduced by Robertson and Seymour [11]) and a parse-tree (as introduced by Wimer [14]) for partial k-trees. We show that a tree decomposition can be constructed in *O(log n)* time on a CRCW PRAM using $O(n^{2k+6})$ processors and a parse-tree in $O(\log^2 n)$ time on a CREW PRAM using the same number of processors. The technique we use involves a recursive decomposition on *feasible centroids* of a graph. Our results improve on a recent result of Bodlaender for constructing a tree decomposition in *O(log n)* time using $O(n^{3k+4})$ processors on a CRCW PRAM.

1. Introduction

Several methodologies have been proposed [2,4,14] in recent years for developing linear-time algorithms for a large class of problems on partial k-trees (also known as partial k-chordal graphs, (k+1)-terminal graphs [15], and graphs of treewidth ≤ k ([11])). All of these methodologies seem to be variations on a common theme. In particular, the methodologies of Wimer [14] and Bodlaender [4] have the following steps:

1. Represent the graph in the form of a 'suitable' tree structure, and
2. Use dynamic programming on this tree structure to solve a problem of interest.

The first step above is known to be computationally hard in general, since Arnborg, Proskurowski and Corneil [1] have shown that it is NP-complete to determine if there exists a k-tree embedding for an arbitrary graph, given k. However, they also showed that a k-tree embedding can be found in $O(n^{k+2})$ time, whenever one exists. The second step requires only linear time for a variety of problems for a fixed value of k, given the tree structure. These problems include finding the stability number, clique number, chromatic number, domination number, etc.

One of the interesting problems in this area is determining whether the above problems can be solved efficiently in parallel, specifically whether there are NC algorithms. We note here that NC algorithms run in polylog time using a polynomial number of processors on 'reasonable models' of parallel computation. The first work in this direction was by He and Yaacov [1,8] who showed that decomposing two-terminal series-parallel graphs (TTSP) and solving various optimization problems can be done efficiently in parallel, following the results of Bern, Lawler, and Wong ([3]). In particular, they showed that the recognition and decomposition of TTSP graphs can be done in $O(\log^2 n+\log m)$ time using *O(n+m)* processors on a EREW PRAM, where *n* (*m*) is the number of vertices (edges) of the graph. Here we show that the problems of recognizing, constructing a tree decomposition and a parse-tree for a partial k-tree are all in NC, for fixed k. Very recently, Bodlaender [5] showed that building a tree decomposition for a partial k-tree can be done in *O(log n)* time using $O(n^{3k+4})$ processors on a CRCW PRAM. Now we point out the important differences between our approaches.

1. Bodlaender constructs a tree decomposition by obtaining 'good' triples (S_i, S_j, R_{ij}) where S_i, S_j are separators of size (k+1) and R_{ij} is the set of vertices v such that v has a path to a vertex in $S_i - S_j$ which avoids S_j and a path to a vertex in $S_j - S_i$ which avoids S_i. Our approach for building the tree decomposition involves finding a k-tree embedding H of G (if one exists) first and constructing the intersection graph representation for H. Finding the embedding also enables us to build a parse-tree for G.

2. Our algorithm for building the tree decomposition uses a factor of $O(n^{k-2})$ fewer processors than Boadlaender's and runs in $O(\log n)$ time on a CRCW PRAM.

The rest of the paper is organized as follows. Section 2 contains the definitions and key results needed for our purposes. The development of results leading to the parallel algorithms is given in Section 3. Section 4 describes the algorithms and the analyses of parallel implementations of the algorithms. Finally some remarks are made in Section 5.

2. Definitions

We shall consider graphs $G=(V,E)$, where V is the vertex set and E is the edge set, as being undirected, connected, without self-loops and multiple edges. If $W \subseteq V$ then we denote by $G[W]$, the graph induced by W in G. The class of k-trees is defined recursively as follows:

1. The complete graph on k vertices is a k-tree.
2. A k-tree H with (n+1) vertices ($n \geq k$) can be constructed from a k-tree G with n vertices by adding a new vertex to G and joining it to all of the vertices of a clique of size k in G. □

Chordal graphs are defined similarly by allowing a new vertex to be joined to all of the vertices of a clique of arbitrary size. A k-chordal graph is a chordal graph in which every clique has a size at most (k+1). A partial k-tree (partial k-chordal graph) G is a subgraph of a k-tree (k-chordal), say H. In this case we say that H is an embedding of G. A vertex v is simplicial if the set of vertices adjacent to v forms a complete subgraph. Sometimes we call a special set of vertices in a graph, the set of terminals of G and we represent the graph by $G=(V,E,T)$ where $T \subseteq V$ is the set of terminals ([12]). If $G_1=(V_1,E_1,T_1)$ and $G_2=(V_2,E_2,T_2)$ then a composition of G_1 and G_2 denoted by $G_1 o G_2 = G = (V,E,T)$ is a graph is obtained by identifying in a 1-1 manner a subset R of the terminals in T_2 with an equal number of terminals S in T_1. Further we require that $G_1[S]$ be isomorphic to $G_2[R]$. In G, we have $V=V_1 U V_2 - R$ and $E=E_1 U E_2$ and $T=T_1$. Partial k-trees can also be defined in the following manner:

1. Each subgraph $H=(V,E,T)$ of K_r, where $r \leq k+1$ is a partial k-tree with $T=V$. These graphs are called the basis graphs.
2. If G_1 and G_2 are partial k-trees then $G_1 o G_2$ is a partial k-tree.
3. All and only those graphs obtained by applying steps 1 and 2 a finite number of times are partial k-trees. (see Wimer [15]). □

A parse-tree H of a partial k-tree G is defined as follows:

1. H is a rooted binary tree in which the root corresponds to G.
2. The leaves of H correspond to basis graphs such that by applying the composition operation 'suitably' bottom up at every non-leaf vertex of H, G can be obtained at the root. □

In other words, a parse-tree H of a partial k-tree G contains the history of the construction of G from a set of basis graphs by repeated composition operations. A tree decomposition of a graph $G=(V,E)$ introduced by Robertson and Seymour [10], is a pair (T,X) where T is a tree and $X=\{A_t : t \text{ in } T\}$ is a family of subsets of V satisfying:

1. $\cup A_t = V$, $t \in T$.
2. For each edge $e=uv$ in E, there exists a t in T such that $u,v \in A_t$ and
3. For t_1,t_2,t_3 in T, if t_2 is on the unique path between t_1 and t_3 in T, then $A_{t_1} \cap A_{t_3} \subseteq A_{t_2}$.

The width of a tree decomposition (T,X) of G is $\max\{|A_t|-1 : t \text{ in } T\}$. The treewidth of G is the minimum width among all tree decompositions of G. In Fig. 1 we give an example of a partial 3-tree G, a 3-tree embedding for G, a tree decomposition and a parse-tree for G. We close this section with a well-known result.

Theorem 2.1. (see Wimer [15], for instance)

The following families of graphs are identical:

1. Partial k-trees,
2. Partial k-chordal graphs, and
3. Graphs having treewidth ≤k.

☐

In the next section we introduce some key notions leading to a separator theorem for partial k-trees.

3. A Separator Theorem for Partial k-Trees

A set C of vertices of a connected graph G is a separator if G-C is disconnected. If C induces a clique in G then we call C a clique-separator. If C is a separator of a graph G with n vertices and every connected component of G-C has size at most $\lfloor n/2 \rfloor$, then we call C a centroid. Further, if C is a centroid and induces a clique in G then C is a clique-centroid. A chordal graph is proper if G is not complete. Now we recall some notions and results from [6]. A maximal clique-separator of a chordal graph is a clique-separator not properly contained in any clique-separator. A maximal clique-centroid is a centroid which is also a maximal clique-separator.

Theorem 3.1. ([6])

For a chordal graph G, a maximal clique-separator is properly contained in a maximal clique S of G if and only if either

a. S is not a separator and has at least two simplicial vertices, or
b. G consists of exactly two maximal cliques (one of which is S) each having precisely one simplicial vertex. ☐

The above result plays a crucial role in the following theorem.

Theorem 3.2. ([6])

Every proper chordal graph contains a maximal clique-centroid.

☐

From the above facts it is clear that we cannot say much about the sizes of the clique-centroids of a chordal graph. However as we shall see soon, for a proper k-tree we can always find clique-centroids of size either k or (k+1). The following lemma is easy to prove.

Lemma 3.3.

Any maximal clique of a proper k-tree G has at most one simplicial vertex.

☐

Now we give one of the main results of this section.

Lemma 3.4.

Every proper k-tree G having n vertices has a clique-centroid of size either k or (k+1).

Proof.

It follows from Theorem 3.2 that G has a maximal clique-centroid, say S. Further it is easy to see that S is either a maximal clique by itself or properly contained in a maximal clique of *G*. If *S* is properly contained in a maximal clique *C* of *G* by Theorem 3.1 and Lemma 3.3 we can infer that *G* has exactly two maximal cliques (one of which is *C*) each having precisely one simplicial vertex. It is obvious then that *G* has a clique-centroid of size k. Note that in this case n=k+2. If *S* is a maximal clique in *G* then clearly the size of *S* is (k+1). In this case we would have $n \geq k+3$.

□

If *S* is a subset of vertices of a graph *G* then clique (*S*) denotes the graph $(S,\{(u,v):u,v \in S, u \neq v\})$. Let *S* be a separator of *G* and $S_1, S_2, ..., S_r$ the connected components of *G-S*. Denote by S^j, $1 \leq j \leq r$, the graph $S_j \cup$ clique(*S*). The sequential algorithm for recognizing a partial k-tree given by Arnborg, Corneil, and Proskurowski ([1]) relies on the following characterization of partial k-trees.

Theorem 3.5. ([1])

A graph *G* of size at least (k+2) is a partial k-tree if and only if there exists a separator *S* of size k such that all the S^j's are partial k-trees.

□

It is clear that a parallel algorithm based on the above theorem may lead to O(n) steps if an arbitrary separator of size k is used at every level of decomposition. In the next section we show that centroids of size k or (k+1) are sufficient for our purposes.

4.NC Algorithms for Parsing and Tree Decomposing Partial k-Trees,for fixed k

A separator *S* of size (k+1) is said to be feasible in a partial k-tree *G* if there exists a k-tree embedding *H* of *G* such that *S* induces a K_{k+1} in *H*. Let *G* be a partial k-tree of size at least (k+3) and *H* an embedding of *G*. Then by applying Lemma 3.4 we can get a clique-centroid for *H* of size (k+1) and hence a feasible centroid of size (k+1) for *G*. The following result brings out the importance of feasible separators.

Lemma 4.1.

Let *G* be a partial k-tree with at least (k+3) vertices. Then a separator *S* of size (k+1) is feasible if and only if all the S^j's are partial k-trees.

Proof.

(Necessity): Let *S* be a feasible separator of size (k+1) in *G*. Then there exists a k-tree embedding *H* of *G* such that *H[S]* is a K_{k+1}. Let *X* be a connected component of *G-S*. It is a fairly straightforward exercise to show that $H[S \cup X]$ is a k-chordal graph and hence a partial k-tree. It follows that $G[X] \cup$ clique(*S*) is a partial k-tree.

(Sufficiency): Let *S* be a separator of size (k+1) in *G* such that the S^j's are partial k-trees. Embed each S^j in a k-tree H^j for every j. Now we can find a perfect elimination ordering for each H^j that eliminates vertices of *S* last. Further all of these perfect elimination orderings can be combined into one (for a k-tree embedding of *G*) that orders vertices of *S* last. Therefore *S* is feasible. □

The following theorem is similar to Theorem 3.5.

Theorem 4.2.

A graph *G* of size at least (k+3) is a partial k-tree if and only if there exists a separator *S* of size (k+1) for *G* such that all the S^j's are partial k-trees.

Proof. Follows from Lemma 4.1 and earlier observations.

□

We know that all graphs of size at most (k+2) with the exception of K_{k+2} are partial k-trees. Hence it is easy to recognize and parse a k-tree having at most (k+2) vertices in parallel in *O(log n)* time. Hence we need to consider only the case when n is at least (k+3).

For a graph *G* with size at least (k+3) our algorithm for recognizing a partial k-tree recursively tries to find a feasible centroid of size (k+1) at every level of decomposition (by examining all centroids at every level). It is clear that *G* is a partial k-tree if and only if there exists a sequence of centroidal decompositions such that there are only proper subgraphs of K_{k+2} (each having a K_{k+1} as a subgraph) left at the bottom of the recursion.Based on the above facts we now describe an algorithm in two parts for recognizing partial k-trees. The first part of the algorithm obtains enough information about the decomposition structure of a graph and the second part makes use of this information efficiently.

Procedure PARTIAL k-TREE (I PART)

```
Input: A graph G of size at least (k+3).
1. Find all separators of G of size (k+1) and obtain the respective sets of connected
   components. Denote the set of all such separators of G by S. Let $T=\{R^j: R \varepsilon S\}$.
2. For every $H \ \varepsilon T$ do the following steps recursively:
    a. Find all the centroids C of size (k+1) for H, if they exist.
    b. Obtain the graphs $C^j$'s for each centroid C.
```

□

Algorithm PARTIAL k-TREE (II PART)

```
1. Invoke procedure PARTIAL k-TREE(I PART).
2. For each centroid Q in S do
       H:=G;
       Carry out steps (a), (b), and (c) recursively
       a. Obtain the graphs $Q^j$'s of H.
       b. if $|Q^j| \geq k+3$ then
              find the centroids P of $Q^j$. If there is no centroid for $Q^j$ then $Q^j$ is not a partial
                k-tree.
          else
              if $Q^j \neq K_{k+2}$ then $Q^j$ is a partial k-tree
              else
                 $Q^j$ is not a partial k-tree.
       c. $H:=Q^j$;
3. if there exists a centroid D of G in S such that every graph $D^j$ is a partial k-tree then G
   is a partial k-tree
       else G is not a partial k-tree
```

□

The correctness of the algorithm follows from Lemma 4.1 and Theorem 4.2. Now we carefully describe the analysis of a fast parallel implementation of the algorithm.

1. There are $O(n^{k+1})$ separators of size (k+1) in a graph. Finding the separators and their respective connected components can be done using $O(n^{k+1}.n^2)$ processors on a CRCW PRAM in $O(\log n)$ time using Shiloach and Vishkin's parallel connectivity algorithm ([12]).

2. Since there are $O(n)$ connected components for every separator of G, the total number of graphs in T (step1 of the I part) is $O(n^{k+2})$, for fixed k. The depth of recursion in step 2 is clearly $O(\log_2 n)$, since centroids divide a graph into pieces at most half its size. In every level of the recursion the following computations are performed.

For every one of $O(n^{k+2})$ graphs, its centroids S and the corresponding S^j's are found. By an exhaustive search procedure in parallel the S^j's can be found in constant time using $O(n^{k+2}.n^{k+1}.n^3)$ processors making use of the information obtained in step1. Note that we do not have to use any algorithm that finds the connected components again here. The details of this would be given in the final paper. Hence the I part of the algorithm can be implemented in parallel using $O(n^{2k+6})$ processors and in $O(\log n)$ time on a CRCW PRAM.

3. The step 2 of the II part of the algorithm has $O(n^{k+1})$ branches examining the feasibility of the centroids in S. In each of these branches steps (a) and (b) can be done in constant time by

accessing the information obtained by the I part of the algorithm using $O(n^{k+2})$ processors at every level. Again the number of levels in every branch is $O(\log n)$. Hence step 2 uses $O(n^{2k+3})$ processors and runs in $O(\log n)$ time.

□

In all the recognition of a partial k-tree can be done using $O(n^{2k+6})$ processors in $O(\log n)$ time on a CRCW PRAM. As we noted earlier a k-tree embedding is easily obtained from a correct sequence of decompositions without requiring additional computational resources. Now we turn our attention to finding a tree decomposition of a partial k-tree. Our approach to this problem depends on the following result which we note without proof.

Lemma 4.3.

If G is a chordal graph then a tree decomposition (T,X) for G is the intersection graph representation for G in terms of the maximal cliques of G. In this representation the vertices of T correspond to the maximal cliques of G, and $X=\{A_t : t \text{ in } T \text{ and } A_t \text{ is a maximal clique of } G\}$ such that if $S_v = \{t : v \varepsilon A_t, A_t \varepsilon X\}$, $v \varepsilon G$ then S_v induces a subtree of T.

□

The intersection graph representation of a chordal graph can be obtained in parallel using $O(n^7)$ processors in $O(\log^2 n)$ time on a CREW PRAM as shown by Naor, Naor and Schaffer [10]. Their approach involves recursion on the centroids of T knowing only the nodes of the tree. Our parallel recognition algorithm for partial k-trees can be used to find the maximal clique-centroids of a k-tree embedding of G (assuming G is a partial k-tree) at various levels of decomposition. The time-complexity of Naor, Naor, and Schaffer's algorithm is dominated by n connectivity operations on a CREW PRAM. In a CRCW PRAM we can, by adopting the same technique find the tree decomposition in $O(\log n)$ time using no extra resources.

Finally we describe how to get a parse-tree of a partial k-tree from its tree decomposition. Our approach is similar in spirit to that of Naor, Naor and Schaffer for finding the perfect elimination ordering of a chordal graph. We would, whenever necessary, closely follow their terminology. We assume that for a partial k-tree G the tree decomposition (T,X) obtained by our method is already known. A terminal branch of T is a path consisting of nodes of T with degree two or less and containing a leaf. To obtain a parse-tree for a partial k-tree G from its tree decomposition we basically do the following steps recursively.

1. Process the nodes in every terminal branch.
2. Delete all the terminal branches.

We note that deleting the terminal branches results in a new tree with half as many leaves as there are in the old tree ([10]). Hence the depth of recursion is $O(\log n)$. The processing of a terminal branch consists of creating a 'binary-tree template'. Intuitively, the binary-tree template would be the parse-tree for that branch. We give an example of a binary-tree template of a terminal branch in Fig. 2.

Now we sketch the algorithm for building a parse-tree of a partial k-tree.

Algorithm PARSE-TREE.

1. Find all the terminal branches of the tree T.
2. Generate the binary-tree template for every terminal branch and augment it with information about the composition operation.
3. Delete the terminal branches.
4. If the tree T is reduced to a path (or a single node) then merge the templates generated otherwise repeat the above steps.

□

We now illustrate the algorithm with an example in Fig. 3. For convenience we ignore the sets associated with the tree nodes. Here is a quick analysis of a parallel implementation of the algorithm.

1. Finding the terminal branches can be done in *O(log n)* time on a CREW PRAM.
2. Generating the templates can be done in constant time.
3. Merging the templates can be done in constant time with some extra bookkeeping.
4. Since there are *O(log n)* stages (each stage taking *O(log n)*time) the time-complexity is $O(\log^2 n)$ on a CREW PRAM.

□

We observe here that our algorithm for recognizing partial k-trees and building a tree decomposition can be done in $O(\log^2 n)$ time using $O(n^{2k+6})$ processors on a CREW PRAM. This can be done by using the Hirschberg, Chandra, and Sarwate ([9]) algorithm for carrying out step 1 of PARTIAL k-TREE (I PART). Hence a parse-tree can be built using the same amount of resources on a CREW PRAM.

5. Remarks

Using the parse-tree of a partial k-tree it seems possible to obtain NC algorithms for various optimization problems which are solvable under Wimer's methodology. This would be a natural parallel extension to his methodology for sequential algorithms. Bodlaender's approach to solving problems using a tree decomposition blows up the treewidth by a factor 3. We believe that our approach would result in simpler and more efficient algorithms.

References

[1] Arnborg, S., D. Corneil, and A. Proskurowski, Complexity of finding embeddings in a k-tree, Siam J. Alg. Disc. Meth., 8, No.2, (1987), pp.277-284.

[2] Arnborg, S., A. Proskurowski, Linear-time algorithms for NP-hard problems restricted to partial k-trees, March 1986, Preprint.

[3] Bern, M.W., E.L. Lawler, and A.L., Wong, Why certain subgraph computations require only linear time, in "Proceedings 26th IEEE Found. of Comp. Science, 1985", pp.117-125.

[4] Bodlaender, H.L., Dynamic Programming algorithms on graphs with bounded treewidth, MIT-LCS-TR-394, MIT Laboratory for Comp. Sc., Cambridge, Mass., 1987.

[5] Bodlaender, H.L., NC algorithms for graphs with small treewidth, Tech. Rept., RUU-CS-88-4, Dept. of Comp. Sc., Univ. of Utrecht, Netherlands, Feb. 1988.

[6] Chandrasekharan, N., R. Laskar, and S.S. Iyengar, Maximal clique-separators of chordal graphs, Tech. Report No. 542-87, Dept. of Mathematical Sciences, Clemson University, South Carolina. (to appear in Congressus Numerantium, 1987.

[7] He, X., and Y. Yaacov, Parallel recognition and decompostion of two terminal series-parallel graphs, Information and Computation, 74, (1987).

[8] He, X., and Y. Yaacov, Binary tree algebraic computations and parallel algorithms for simple graphs, Journal of Algorithms, 8, (1987).

[9] Hirschberg, D.S., A.K. Chandra, and D.V. Sarwate, Computing connected components on parallel computers, CACM, 22 (1979), pp.461-464.

[10] Naor, J., M. Naor, and A.A. Schaffer, Fast Parallel algorithms for chordal graphs, Proc. STOC, NY city, (1987).

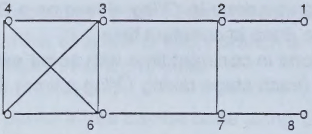
[11] Robertson, N., and P. Seymour, Graph Minors II, Algorithmic aspects of treewidth, Journal of Algorithms, 7, (1986) pp.309-322.

[12] Seese, D., Tree-partite graphs and the Complexity of algorithms, preprint P-MATH 08/86, Akademie der Wissenschaffen der DDR, karl-Weierstrass-Institut fur Mathematik, Berlin, 1986.

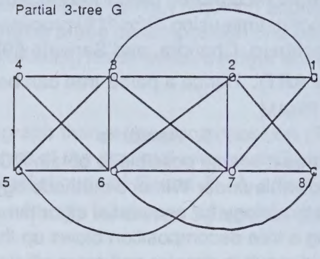
[13] Shiloach, Y., and U. Vishkin, An O(log n) parallel connectivity algorithm, Journal of Algorithms, (1982), pp.57-67.

[14] Wimer, T.V., S.T.Hedetniemi, and R. Laskar, A Methodology for constructing linear graph algorithms, Congressus Numerantium, 50, (1985), pp.43-60.

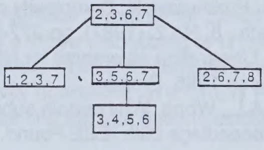
[15] Wimer, T.V., Linear Algorithms on k-Terminal Graphs, Ph.D. Thesis, Dept. of Computer Science, Clemson University, Aug. 1987.



Partial 3-tree G



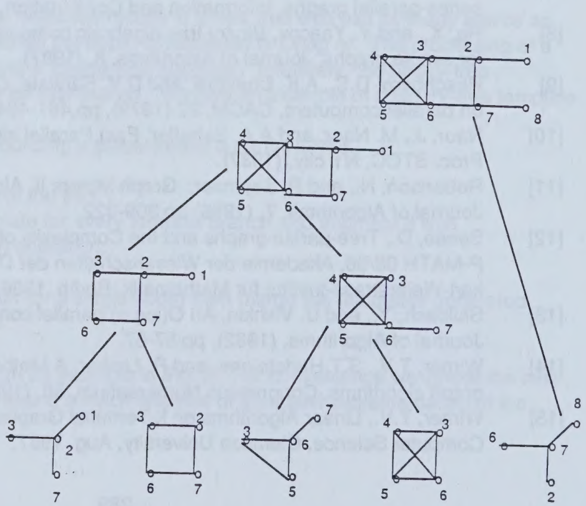
A 3-tree embedding of G


2,3,6,7
1,2,3,7
3,5,6,7
2,6,7,8
3,4,5,6

A tree decomposition of G

Fig. 1

Fig. 1 (continued) A parse-tree for G



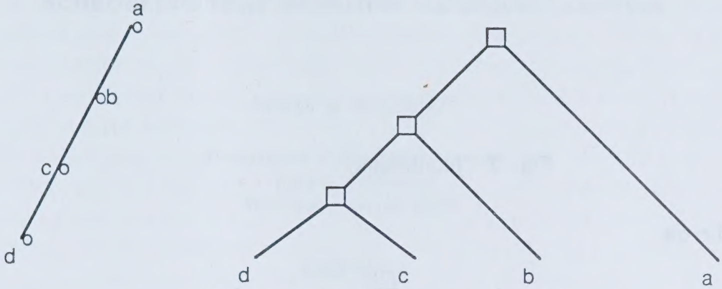
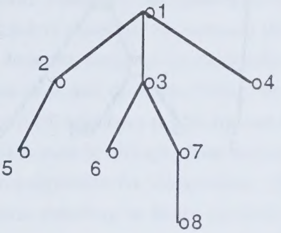


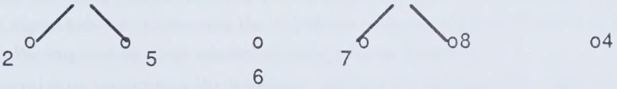
Fig. 2



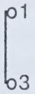
Tree Decomposition of a graph

Terminal branches : {2,5} , {6}, {7,8}, {4}

The templates



Reduced tree after deleting the terminal branches



Template

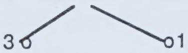
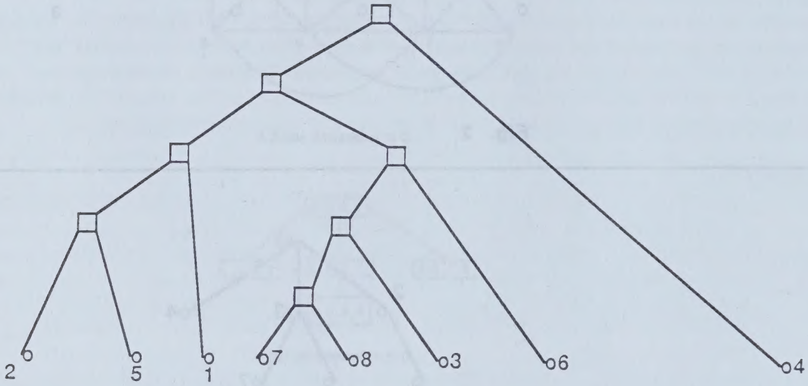


Fig. 3

Fig. 3 (continued)

Merge



AN OPTIMAL PARALLEL ALGORITHM FOR PREEMPTIVE JOB SCHEDULING THAT MINIMIZES MAXIMUM LATENESS

SUSAN H. RODGER †

Department of Computer Sciences
Purdue University
West Lafayette, IN 47907

ABSTRACT

In this paper we present an optimal parallel algorithm that schedules jobs for execution on a single processor machine with preemption. Each job is described by a release time, a deadline and a processing time. Using the CREW PRAM model, our algorithm schedules n jobs for execution and minimizes the maximum lateness of the jobs. Our algorithm runs in $O(\log n)$ time and uses $O(n)$ processors. The technique we use can be applied to other job scheduling problems, improving their results.

1. INTRODUCTION

In this paper we present an optimal parallel algorithm that preemptively schedules jobs for execution on a single processor machine. Each job is described by a release time, a deadline and a processing time. A job is considered late if it does not complete by its deadline and its lateness is defined as the difference between its completion time and deadline. Using the CREW (concurrent reads, exclusive writes) PRAM model, our algorithm schedules n jobs for execution and minimizes the maximum lateness of the jobs. Our algorithm runs in $O(\log n)$ time and uses $O(n)$ processors, which is the first optimal processor-time product algorithm for this problem. The technique we use can be applied to other job scheduling problems, resulting in faster parallel algorithms for these problems.

There are efficient sequential algorithms for many scheduling problems, but few parallel algorithms for these problems. Sequential job scheduling algorithms that minimize the maximum lateness have been studied extensively. If preemption is allowed, scheduling jobs to run on one machine is solvable in polynomial time [13]. If preemption is not allowed, scheduling jobs to run on one machine is NP-hard [3], but polynomial time algorithms exist if all release times are equal [14], all deadlines are equal [14], or all processing times are equal [11] [18]. Recently, parallel algorithms for job scheduling problems are being studied ([5], [6], [7], [8], [12], and [15]). In particular, Dekel and Sahni [5] give parallel algorithms for minimizing the maximum lateness of jobs scheduled to run on m identical machines. We improve on their results obtaining faster algorithms.

This paper concentrates on minimizing the maximum lateness for the case when jobs are scheduled to run on one machine and preemption is allowed. For this problem, Dekel and Sahni [5] give a parallel algorithm that is based on parallel divide-and-conquer. The bottleneck in their algorithm is that it takes $O(\log n)$ time to combine two subproblems. There are $\log n$ levels of subproblems, so their algorithm takes $O((\log n)^2)$ time. Our algorithm, the Preemptive Minimize Maximum Lateness algorithm, improves the running time by pipelining the subproblems. It consists of two phases. In phase one we apply the pipeline merge technique of Cole [4] to the first part of Dekel and Sahni's

---

† This research was partially supported by the Office of Naval Research on contract N 00014-86-K-0689.

algorithm. Subproblems on all levels calculate their solutions at the same time, so phase one completes in $O(\log n)$ time. The difficulty in pipelining the subproblems lies in determining preempted jobs at intermediate steps, since only a sample of the final solution set is available. At the intermediate steps, the preempted job is estimated. As the algorithm proceeds, the estimate improves and it converges on the exact job when all the jobs in a subproblem are available.

In phase two of our algorithm, we cannot directly apply the pipeline merge technique to the second part of Dekel and Sahni's algorithm. Instead, we design a new algorithm for this phase to which we can then apply Cole's pipeline merge technique. Dekel and Sahni's algorithm runs in $O((\log n)^2)$ time and uses $O(n)$ processors. Our algorithm reduces their running time by $O(\log n)$ while using the same number of processors. Using an adversary argument [10], this problem has a lower bound $\Omega(n \log n)$, so our algorithm has an optimal processor-time product.

This paper is organized as follows. Section 2 reviews previous work that is relevant to our algorithm. Section 3 presents the Preemptive Minimize Maximum Lateness algorithm. Section 4 applies our approach to other job scheduling problems, improving their running times.

2. DEFINITIONS AND PREVIOUS RESULTS

In this section we first define the scheduling problem for minimizing the maximum lateness, and then we review phase one of Dekel and Sahni's [5] parallel algorithm.

Allowing preemption, we want to schedule n jobs to be processed on one machine so that we minimize the maximum lateness of the jobs. Each job i has an associated release time r_i, deadline d_i and processing time p_i. A job cannot start earlier than its release time and it should finish by its deadline. By allowing preemption, jobs can run in time slices, with the sum of the time slices equaling the job's processing time. A job is late if it completes after its deadline. Let c_i be the completion time for job i, then the lateness of job i is defined by $c_i - d_i$. We want to schedule the jobs so that we minimize the maximum $c_i - d_i$ over all jobs i.

2.1. Dekel And Sahni's Parallel Algorithm

Using the EREW (exclusive read, exclusive write) PRAM model, Dekel and Sahni [5] give a parallel solution to the preemptive minimize maximum lateness problem. Dekel and Sahni's algorithm uses parallel divide-and-conquer, dividing the problem into subproblems, solving the subproblems in parallel, and then combining their results. A complete binary tree is used to illustrate the order the subproblems are combined. The jobs are ordered by nondecreasing release times and divided into groups that have jobs with the same release time. The groups are assigned to the leaves of the binary tree, one group per leaf, in the release time order.

An interval is associated with each node in the tree. This interval is based on release times and it represents available time for scheduling jobs. If V is a leaf node that contains jobs with release time r_i and the leaf node to its right contains jobs with release time r_j, then the interval $[r_i, r_j)$ is associated with V. We can schedule some of the jobs in V to run during this interval such that the total time for processing the scheduled jobs is less than or equal to $r_j - r_i$. The rightmost leaf, which has release time $r_{n'}$, represents the interval $[r_{n'}, \infty)$. If V is an internal node, it is associated with the interval $[r_i, r_j)$ where $[r_i, r_k)$ is the interval associated with V's left child and $[r_k, r_j)$ is the interval associated with V's right child.

Phase one begins by assigning one processor per job and in parallel sorting the n jobs by nondecreasing release times. The jobs are then divided into groups by release times and the groups are assigned to the leaves of the tree. In parallel, the jobs within a group are further sorted by nondecreasing deadlines. Each node V calculates two ordered lists of jobs, $SCHED_V$ and REM_V. $SCHED_V$ is the set of jobs, considering only those jobs in V's subtree, that can be scheduled in the interval associated with V to minimize the maximum lateness. REM_V is the set of remaining jobs from V's subtree that are not in $SCHED_V$. Both $SCHED_V$ and REM_V are in sorted order by deadlines, since their solutions can be calculated quickly in $\log n$ time by merging sets sorted by deadlines. The calculation does not produce the sets in their scheduled order. If V is a leaf node then all its jobs have the same release time r_i and the jobs are sorted by nondecreasing deadlines. The interval associated with V is $[r_i, r_j)$. Let $j_1, j_2, \ldots j_t$ be the jobs in V in sorted order. The jobs in V are split into the two sets $SCHED_V$ and REM_V. The partial sums of the processing times are calculated to determine where this split occurs. Let z be the index such that

$$\sum_{h=1}^{z} p_{j_h} \leq r_j - r_i \quad and \quad \sum_{h=1}^{z+1} p_{j_h} > r_j - r_i$$

The jobs $j_1, j_2, \ldots j_z$ are placed into $SCHED_V$. If the first summation does not equal $r_j - r_i$ and job j_{z+1} exists, then j_{z+1} is split into two new jobs. Its processing time is split so that one of the new jobs can be added to $SCHED_V$, resulting in the sum of the processing times of jobs in $SCHED_V$ equal to $r_j - r_i$. The other new job and all the remaining jobs not chosen are placed into REM_V.

If V is an internal node, then $SCHED_V$ and REM_V are calculated using V's left child and right child, $left(V)$ and $right(V)$. V is associated with the interval $[r_i, r_j)$, $left(V)$ with the interval $[r_i, r_k)$ and $right(V)$ with the interval $[r_k, r_j)$. Assume $SCHED_{left(V)}$, $REM_{left(V)}$, $SCHED_{right(V)}$, and $REM_{right(V)}$ have already been calculated. Every job in $right(V)$ has release time r_z such that $r_z \geq r_k$, so $REM_{right(V)} \subset REM_V$. Similarly, $SCHED_{left(V)} \subset SCHED_V$. Of the jobs in $SCHED_{right(V)}$ and $REM_{left(V)}$, choose the jobs with minimum deadlines that use at most $r_j - r_k$ total processing time. This is done as follows. Let $W = SCHED_{right(V)} \cup REM_{left(V)}$ where $\cup$ is a merge operation in order of nondecreasing deadlines. Calculate the partial sums of the processing times in W and using these split W into the two sets $SCHED_W$ and REM_W. W is split in the same manner the leaf nodes were split, so that the sum of the processing times of the jobs in $SCHED_W$ is less than or equal to $r_j - r_k$. This may involve splitting one job into two new jobs as before. REM_W will contain the remaining jobs. Then $SCHED_V = SCHED_{left(V)} \cup SCHED_W$ and $REM_V = REM_W \cup REM_{right(V)}$. $SCHED_V$ will contain the set of jobs that minimize the maximum lateness for V's interval, considering only the jobs in V's subtree. If $ROOT$ is the root of the tree, $SCHED_{ROOT}$ will contain all the jobs in nondecreasing sorted order by deadlines and REM_{ROOT} will be empty. Note that the jobs in $SCHED_{ROOT}$ are not necessarily in scheduled order, since each $SCHED$ and REM set calculates the jobs in the sets, but never the scheduled order of the jobs. At most one job per node is split into two jobs, so there are $O(n)$ jobs at the root. Another phase that starts at the root and finishes at the leaves obtains the schedule.

3. A NEW ALGORITHM FOR MINIMIZING THE MAXIMUM LATENESS

In this section we present the Preemptive Minimize Maximum Lateness algorithm. Phase one of our algorithm applies Cole's pipeline merge technique to phase one of Dekel and Sahni's parallel scheduling algorithm, maintaining partial sums of processing times to estimate which jobs are preempted. Phase two uses the information calculated from phase one to design an algorithm composed of merges to which we can apply Cole's parallel merge technique. The merges in phase two are more complicated than the merges in phase one, as phase two merges do not follow the binary tree structure and some lists may merge with several lists at the same time (see [17] for more detail).

3.1. Phase One

Phase one of our algorithm calculates the set of jobs that can be scheduled, but does not produce the jobs in their scheduled order. This problem is divided into $\log n$ levels of subproblems. Each subproblem has a set of jobs and a time interval in which to schedule the jobs. For each subproblem, phase one can quickly calculate the set of jobs that can be scheduled to run in its associated interval, because the jobs are in sorted order by deadlines. Solutions are calculated by merging and splitting ordered lists. Furthermore, the algorithm is speeded up by pipelining the jobs through the subproblems, calculating the solutions to all subproblems at the same time. Phase two uses the solutions of the subproblems in phase one to calculate the schedule of the jobs.

Phase one begins by sorting in parallel the n jobs by release times, using Cole's [4] parallel merge sort. For those jobs with identical release times, it further sorts these jobs by deadlines. We assign the jobs in their order to the leaves of a binary tree, assigning *one* job per leaf. One processor is assigned to each job and it remains with the job as it is pipelined to the root. An interval is associated with each node in the tree, as described in section 2.1. We calculate the $SCHED_V$ and REM_V sets as defined in section 2.1, but we calculate them using a pipeline. For each job i, e_i is the estimated partial sum of processing times. Initially, $e_i = p_i$, the processing time for job i. If V is a leaf node with job i and interval $[r_i, r_{i+1})$, then job i is placed into $SCHED_V$ if $r_{i+1} - r_i \geq e_i$, or it is placed into REM_V if $r_{i+1} - r_i = 0$. Otherwise job i is split into two jobs, i_1 and i_2, such that i_1 has $p_{i_1} = r_{i+1} - r_i$ and i_2 has $p_{i_2} = p_i - p_{i_1}$. Then i_1 is placed into $SCHED_V$ and i_2 is placed into REM_V.

The algorithm runs in stages. At each stage, operations are performed at each node V in the tree. Let $SCHED_V(s)$ be the current solution set of jobs at node V in stage s. V is *external* if its solution sets contain all the jobs in its subtree. If V is not external, then it contains samples of its final collection of jobs. SAMP($SCHED_V(s)$) will be the sample list of $SCHED_V$ at stage s. If V is not external, SAMP($SCHED_V(s)$) equals every fourth job of $SCHED_V(s)$. If V becomes external at stage s, then SAMP($SCHED_V(s)$) equals every fourth job of $SCHED_V(s)$, SAMP($SCHED_V(s+1)$) equals every second job of $SCHED_V(s)$ and SAMP($SCHED_V(s+2)$) $= SCHED_V(s)$. Note that there is no need to calculate SAMP($SCHED_V(z)$) for $z > s+2$ as the parent of V now has its complete list. When $SCHED_V(s)$ becomes external at stage s, it is equivalent to the list $SCHED_V$ in Dekel and Sahni's algorithm.

To handle the splitting of lists, we create an *intermediate* level between every level in the tree. Levels from the original tree will be called *regular* levels. Let V be a node in the original tree with left child $left(V)$ and right child $right(V)$. On the intermediate level between V and its children, there will be three nodes W, $int\_left(V)$ and $int\_right(V)$. $W(s)$ will be estimated by samples sent up from $REM_{left(V)}(s-1)$ and $SCHED_{right(V)}(s-1)$ and then $W(s)$ will be split into $SCHED_W(s)$ and $REM_W(s)$. $Int\_left(V)$ will receive sample lists from $SCHED_{left(V)}(s-1)$ and $int\_right(V)$ will receive sample lists from $REM_{right(V)}(s-1)$. Finally, $SCHED_V(s)$ and $REM_V(s)$ can be calculated. Figure 1 shows the flow between two regular levels and the intermediate level between them. Note that $int\_left(V)$ and $int\_right(V)$ are not necessary. Instead of these two nodes, smaller samples of $SCHED_{left(V)}$ and $REM_{right(V)}$ could be defined, and SAMP($SCHED_{left(V)}(s-1)$)

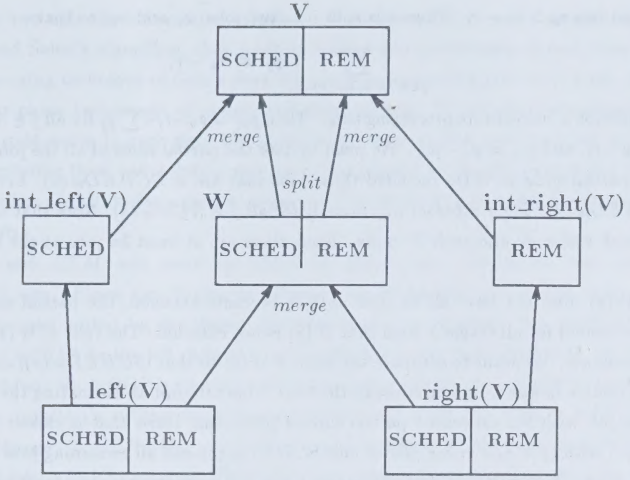


FIGURE 1

would be sent to $SCHED_V(s)$ and SAMP($REM_{right(V)}(s-1)$) would be sent to $REM_V(s)$. We use $int\_left(V)$ and $int\_right(V)$ so that all the sample lists are defined the same, thus making it easier to describe the algorithms and the proofs.

Here is a description of one stage in the algorithm. Let V, $left(V)$, $right(V)$ and W be defined as above. The first step consists of calculating in parallel SAMP($SCHED_V(s-1)$) and SAMP($REM_V(s-1)$) for all nodes V. In the second step, for all nodes of type V, W, $int\_left(V)$ and $int\_right(V)$ calculate in parallel the following (we abbreviate $left$ by l and $right$ by r): $SCHED_V(s) = \text{SAMP}(SCHED_W(s-1)) \cup \text{SAMP}(SCHED_{int\_l(V)}(s-1))$, $SCHED_{int\_l(V)}(s) = \text{SAMP}(SCHED_{l(V)}(s-1))$, $REM_V(s) = \text{SAMP}(REM_W(s-1)) \cup \text{SAMP}(REM_{int\_r(V)}(s-1))$, $REM_{int\_r(V)}(s) = \text{SAMP}(REM_{r(V)}(s-1))$, and $W(s) = \text{SAMP}(REM_{l(V)}(s-1)) \cup \text{SAMP}(SCHED_{r(V)}(s-1))$. The third step consists of splitting $W(s)$ into $SCHED_W(s)$ and $REM_W(s)$ for all W in parallel.

We give more detail on the sample lists and then explain how the split is calculated. As soon as a $SCHED$ or REM list receives its first few jobs, it will approximately double in size at each stage following, becoming external when it has received all its jobs. The jobs start in the leaves of the tree with 0 or 1 jobs in each $SCHED_V$ and REM_V list, for V a leaf node. The jobs are pipelined up the tree level by level in sample lists. Unlike Cole's algorithm, the $SCHED_V$ and REM_V lists can be different sizes throughout the tree. The larger of the two will possibly receive jobs at an earlier stage. Both lists on receiving jobs approximately double in size from one stage to the next, with both lists becoming external (or complete) at the same time.

When $W(s)$ is external, splitting $W(s)$ into $SCHED_W(s)$ and $REM_W(s)$ is easy. Let $[r_i, r_k)$ be the interval associated with $W(s)$, then $SCHED_W(s)$ can choose jobs from $W(s)$ whose partial sum of processing times is at most $r_k - r_i$. When $W(s)$ is external we can calculate which job should be preempted and then split that job into two jobs. All jobs $j \in W(s)$ with $e_j \leq r_k - r_i$ are placed into $SCHED_W(s)$. If there is no job j with $e_j = r_k - r_i$, then let x be the job with the smallest deadline that has $e_x > r_k - r_i$. Then x is split into two jobs, x_1 and x_2, so that

$$\sum_{j \in SCHED_W(s) \cup x_1} p_j = r_k - r_i$$

When we split job x, we split its processing time. Then $p_{x_1} = r_k - r_i - \sum p_j$ for all $j \in SCHED_W(s) - x_1$, $e_{x_1} = r_k - r_i$, and $p_{x_2} = p_x - p_{x_1}$. We must update the partial sums of all the jobs in $REM_W(s)$ since their partial sums in $W(s)$ included those jobs that are in $SCHED_W(s)$. Let j_L be the last job in $SCHED_W(s)$. Then subtract e_{j_L} from e_j for all $j \in REM_W(s)$. Note that at most one job is split at each leaf node and each W node. Thus, there are at most $3n$ jobs at the root at the end of phase 1.

Since $W(s)$ does not have all its jobs until it becomes external, the partial sums for jobs in $W(s)$ are estimated for all stages s such that $W(s)$ is not external. The split of $W(s)$ at each stage will be an estimate. We want to estimate the split of $W(s)$ so that $|SCHED_W(s)|$ and $|REM_W(s)|$ are at most double in size from one stage to the next. One attempt at estimating the split is to split $W(s)$ at the job with the estimated partial sum of processing times that is closest to $r_k - r_i$. All jobs $j \in W(s)$ with $e_j < r_k - r_i$ are placed into $SCHED_W(s)$ and all remaining jobs are placed into $REM_W(s)$. Since the partial sums of processing times are just estimates we don't need an exact fit into the interval until a node is external. Job splitting will occur only when nodes are external.

One problem with this estimate for the split is that we cannot guarantee that $|SCHED_W(s)|$ and $|REM_W(s)|$ will at most double in size from stage $s-1$ to stage s. Instead of estimating the split in just one position, estimate the split by a range of values. Make two cuts in $W(s)$, a high cut HC and a low cut LC. Let $w_1, w_2, \ldots w_l$ be the jobs in $W(s)$ and let w_m be the job with the largest deadline that has $e_{w_m} < r_k - r_i$. Without loss of generality, w_m is from SAMP$(SCHED_{right(V)}(s-1))$. Let w_j be the first job to w_m's left that came from the other list SAMP$(REM_{left(V)}(s-1))$. Similarly let w_h be the first job to w_m's right that came from the other list SAMP$(REM_{left(V)}(s-1))$. Then the jobs $w_{j+1}, \ldots, w_m, \ldots, w_{h-1}$ all came from the same list SAMP$(SCHED_{right(V)}(s-1))$ and they define the *region of uncertainty*. The exact split falls somewhere in this region. The low cut LC will be between jobs w_j and w_{j+1} and the high cut HC will be between jobs w_{h-1} and w_h. All jobs in the region of uncertainty will be duplicated and placed into both $SCHED_W(s)$ and $REM_W(s)$. In later stages as the interval of uncertainty shrinks, the LC and HC approach each other and will converge into one cut by the time this node becomes external. So $SCHED_W(s) = w_1, w_2, \ldots w_{h-1}$ and $REM_W(s) = w_{j+1}, \ldots w_l$.

Theorem 1: Phase one of our algorithm that preemptively schedules jobs to run on one machine while minimizing the maximum lateness of the jobs, runs in $O(\log n)$ time using $O(n)$ processors.

3.2. Phase Two

Phase two of our algorithm calculates the schedule of jobs, using results from phase one. In phase one, each node in the tree is given the set of jobs *in its subtree* and a time interval. Using these, it calculates the solution set of jobs that can be scheduled in its associated interval. Phase two updates the solution sets at nodes by considering *all* the jobs in the tree when finding the schedule for a node's interval. These new solution sets are called NEW_$SCHED_V$. Now, the nodes have the exact set of jobs to schedule in their associated intervals, but the jobs are still not in scheduled order. However, the schedule that minimizes the maximum lateness can be constructed by concatenating the NEW_$SCHED_V$ lists in the leaves of the tree.

Our phase two uses the information calculated from phase one to avoid extractions. In phase two of Dekel and Sahni's algorithm, they perform merges and extractions at each node on a level. The parallel merging technique of Cole's does not apply to extracting jobs from a set. Eliminating extractions, our phase two consist of merges and splits of lists. To calculate new solution sets, we can bypass all right nodes (except right leaf nodes which are a special case) and just work with the left nodes, calculating their new solution sets using merges and splits. We discuss the beginning of phase two, give the algorithm for a general node and then show how to apply Cole's parallel merge to our algorithm.

$SCHED$ and REM sets were calculated in phase one. In phase two, we calculate $\mathrm{NEW\_}SCHED$ sets. Phase two begins as follows. Let V be the root of the tree. $SCHED_V$ contains all the jobs sorted by deadlines, the solution for its interval. So, $\mathrm{NEW\_}SCHED_V = SCHED_V$. Let $left(V)$ be the left child of V and $right(V)$ be the right child of V. $SCHED_{left(V)}$ contains the solution for $left(V)$'s interval when considering all the jobs in $left(V)$'s subtree. Since all the jobs in $right(V)$'s subtree have larger release times than the jobs in $left(V)$'s subtree, $\mathrm{NEW\_}SCHED_{left(V)} = SCHED_{left(V)}$. $REM_{left(V)}$ contains all the jobs in $left(V)$'s subtree that did not fit into $left(V)$'s interval. These jobs have release times less than any job in $right(V)$'s subtree, so they should be considered in the new solution sets in $right(V)$.

We do not need to calculate $\mathrm{NEW\_}SCHED_{right(V)}$ to update the solution sets in $right(V)$'s subtree. By merging $SCHED_{left(right(V))}$ and $REM_{left(V)}$ together, $\mathrm{NEW\_}SCHED_{left(right(V))}$ can be calculated by chosing the jobs with minimum deadlines whose sum of processing times fits into $left(right(V))$'s interval. This will consist of one merge, $SCHED_{left(right(V))} \cup REM_{left(V)}$, and then one split, splitting the list so that $\sum p_i = left(right(V))$'s interval, for all $i \in \mathrm{NEW\_}SCHED_{left(right(V))}$. Thus, we can bypass $\mathrm{NEW\_}SCHED_{right(V)}$ in calculating $\mathrm{NEW\_}SCHED_{left(right(V))}$.

$Left(right(V))$'s children consider some of the jobs from $REM_{left(V)}$ in their $\mathrm{NEW\_}SCHED$ sets. (We abbreviate *left* by l and *right* by r.) $\mathrm{NEW\_}SCHED_{l(r(V))}$ should be divided into new solutions for $\mathrm{NEW\_}SCHED_{l(l(r(V)))}$ and $\mathrm{NEW\_}SCHED_{r(l(r(V)))}$. $\mathrm{NEW\_}SCHED_{l(l(r(V)))}$ is formed by considering jobs in $SCHED_{l(l(r(V)))}$ and those jobs in $\mathrm{NEW\_}SCHED_{l(r(V))}$ that have a release time less than $l(r(V))$'s interval, which are some of the jobs from $REM_{l(v)}$. When $REM_{l(V)}$ and $SCHED_{l(r(V))}$ were merged together and then split, $REM_{l(V)}$ is really split in half with those jobs in the left half, call these $REM^{(1)}_{l(V)}$, going into $\mathrm{NEW\_}SCHED_{l(r(V))}$. $REM^{(1)}_{l(V)}$ is the set of jobs that have release time less than $l(l(r(V)))$'s interval and which we want to consider for $\mathrm{NEW\_}SCHED_{l(l(r(V)))}$. Thus to form $\mathrm{NEW\_}SCHED_{l(l(r(V)))}$, merge together $SCHED_{l(l(r(V)))}$ and $REM^{(1)}_{l(V)}$, and then split the list to fit into $l(l(r(V)))$'s interval. We do not need to extract any jobs from $\mathrm{NEW\_}SCHED_{l(r(V))}$. Figure 2 shows the flow of jobs in phase two starting with $left(V)$. The abbreviations used in this figure are S for $SCHED$, R for REM, and NS for $\mathrm{NEW\_}SCHED$.

Here is the algorithm for a general node in phase two. Let $left(V)$ be a node in the tree with interval $[r_i, r_j)$ and lists $SCHED_{left(V)}$ and $REM_{left(V)}$ calculated from phase one. Assume that jobs in the tree to the left of $left(V)$'s subtree that have not been selected yet are merged together with $SCHED_{left(V)}$ and then split, the left half forming $\mathrm{NEW\_}SCHED_{left(V)}$ and the right half merging with $REM_{left(V)}$ to form $REM^{(0)}_{left(V)}$.

Let $right(V)$ be the right sibling of $left(V)$. $Right(V)$ is a right node so we will ignore it, but note that the set of jobs in $REM^{(0)}_{left(v)}$ should be considered in $right(V)$'s subtree. Let $left(V)_1$ be $right(V)$'s left child, $left(V)_2$ be $left(V)_1$'s left child, $left(V)_3$ be $left(V)_2$'s left child, etc.

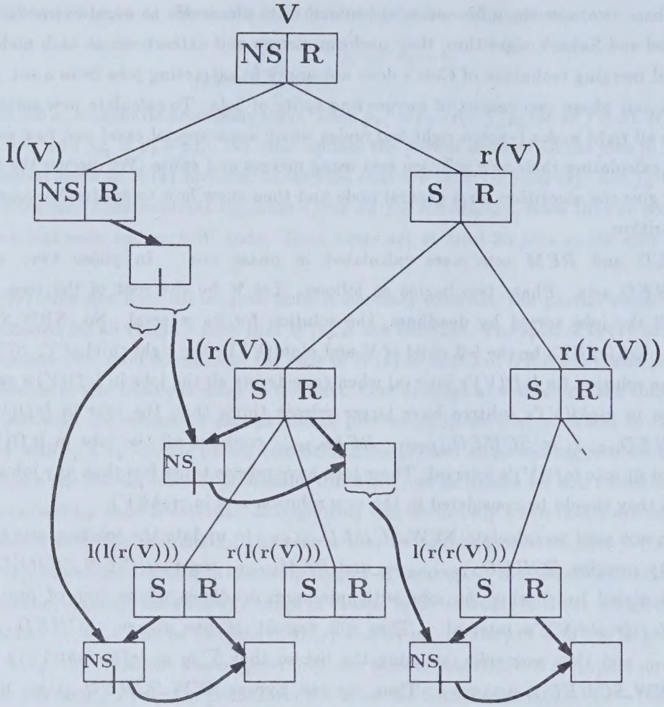


FIGURE 2

To form NEW_$SCHED_{left(V)_i}$, merge $SCHED_{left(V)_i}$ and $REM^{(i-1)}_{left(v)}$ to form NEW_$left(V)_i$. Split NEW_$left(V)_i$ into NEW_$SCHED_{left(V)_i}$ and a right half which is merged with $REM_{left(V)_i}$ to form $REM^{(0)}_{left(V)_i}$. When NEW_$left(V)_i$ is split, use this split to split $REM^{(i-1)}_{left(V)}$ so that $REM^{(i)}_{left(V)} = REM^{(i-1)}_{left(V)} \cap$ NEW_$SCHED_{left(V)_i}$.

That is the general algorithm for phase two. We now organize phase two so that we can apply the pipeline merge technique used in phase one. An intermediate level is added between every two levels in the tree. From the description of the general algorithm, there are two merges and one split for each left node in the original tree. At node $left(V)_i$, $SCHED_{left(V)_i}$ and $REM^{(i-1)}_{left(V)}$ are merged together into NEW_$left(V)_i$ and then split, with the left half of the split forming NEW_$SCHED_{left(V)_i}$. The right half of the split is then merged with $REM_{left(V)_i}$ to form $REM^{(0)}_{left(V)_i}$. So NEW_$left(V)_i$ is formed and split at the level where $left(V)_i$ resides and $REM^{(0)}_{left(V)_i}$ is formed on the intermediate level directly below $left(V)_i$'s level. Similar to phase one, we have alternating levels performing merges and splits on one level and just merges on the next level.

Theorem 2: Phase two of our algorithm that preemptively schedules jobs to run on one machine while minimizing the maximum lateness of the jobs, runs in $O(\log n)$ time using $O(n)$ processors.

4. APPLICATIONS

The technique of pipelining the merges of subproblems to improve an algorithm's running time can be applied to other job scheduling problems. In this section we apply this technique to get faster parallel algorithms for four job scheduling problems. Dekel and Sahni [5] [6] (first, third and fourth problems) and Dolev et al. [8] (second problem) show that these problems are in NC and give parallel algorithms for them that run in $O((\log n)^2)$ time and use $O(n)$ processors. Our algorithms are similar to the algorithm in section 3 and run in $O(\log n)$ time using $O(n)$ processors.

In the first job scheduling problem we are given n jobs with integer release times, deadlines and unit processing times. We want to minimize the maximum lateness when scheduling the jobs to run on m identical machines. If $m = 1$, the algorithm from section 3 can be used to solve this problem. There is no preemption and thus no job splitting due to the integer release times, so this is a simpler version of the problem in section 3. When $m > 1$, define interval $[r_i, r_j)$ to contain at most $m(r_j - r_i)$ jobs and again apply the algorithm from section 3. In both cases, our algorithm runs in $O(\log n)$ time using $O(n)$ processors. Atallah *et al.* [2] also give a parallel algorithm for this problem that runs in $O(\log n)$ time and uses $O(n)$ processors. This is not necessarily optimal as Frederickson [9] gives a linear sequential algorithm for this problem.

Corollary 1: Defining jobs by integer release times, deadlines and unit processing times, there is an algorithm for the job scheduling problem of minimizing the maximum lateness of jobs that schedules the jobs to run on m machines in $O(\log n)$ time using $O(n)$ processors.

In the second job scheduling problem we are given n jobs with unit processing times and a set of precedence constraints in the form of outtrees. We want to find an optimal schedule to run on m identical machines. Dolev et al. [8] give an EREW parallel algorithm for this problem which runs in $O((\log n)^2)$ time and uses $O(n)$ processors. They use Dekel and Sahni's [7] parallel matching algorithm as a subroutine for finding an earliest deadline schedule. Instead, using the algorithm in Corollary 1 as their subroutine results in an $O(\log n)$ time CREW algorithm using $O(n)$ processors.

Corollary 2: Defining jobs by unit processing times and outtree precedence constraints, there is an algorithm for scheduling the jobs to run on m machines in $O(\log n)$ time using $O(n)$ processors.

In the third job scheduling problem we are given n jobs with the same release times ($r_i = 0$), integer deadlines, unit processing times, and a weight. A job is tardy if it completes after its deadline. We want to minimize the sum of the weights of tardy jobs when scheduling the jobs to run on one machine. In this case, sort the jobs by deadlines, define the intervals using the deadlines and place the jobs in the leaves of the binary tree. The algorithm is similar to phase 1 of the algorithm from section 3, at each node choosing the jobs with largest weights to fit into the intervals. At the root r, $SCHED_r$ contains the nontardy jobs and REM_r contains the tardy jobs. The schedule that minimizes the sum of the weights of tardy jobs is obtained by sorting the jobs in $SCHED_r$ by nonincreasing deadlines and then appending the REM_r jobs to the end of the schedule. This algorithm runs in $O(\log n)$ time using $O(n)$ processors.

Corollary 3: Defining jobs by release times equal to zero, integer deadlines, unit processing times, and weights, there is an algorithm for the job scheduling problem of minimizing the sum of the weights of tardy jobs that schedules the jobs to run on one machine in $O(\log n)$ time using $O(n)$ processors.

In the fourth job scheduling problem we are given n jobs with the same release times ($r_i = 0$), arbitrary deadlines and unit processing times. We want to minimize the number of tardy jobs when scheduling the jobs to run on one machine. Monma [16] gives a sequential algorithm for solving this problem that runs in $O(n)$ time. An algorithm similar to the algorithm in section 3 gives an $O(\log n)$ time algorithm that uses $O(n)$ processors.

Corollary 4: Defining jobs by release times equal to zero, integer deadlines, and unit processing times, there is an algorithm for the job scheduling problem of minimizing the number of tardy jobs that schedules the jobs to run on one machine in $O(\log n)$ time using $O(n)$ processors.

Acknowledgements: I would like to thank Greg Frederickson and Dah Jyh Guan for discussions and comments.

5. REFERENCES

[1] Atallah, M. J., Cole, R., and Goodrich, M. T., Cascading Divide-and-Conquer: A Technique for Designing Parallel Algorithms, *Proc. of the 28th Annual Symp. on Foundations of Computer Science* (1987), 151–160.

[2] Atallah, M. J., Goodrich, M. T., and Kosaraju, S. R., On the Parallel Complexity of Evaluating Some Sequences for Set Manipulation Operations, *3rd International Workshop on Parallel Computation and VLSI Theory (AWOC)* (1988).

[3] Brucker, P., Rinnooy Kan, A. H. G., and Lenstra, J. K., Complexity of Machine Scheduling Problems, *Ann. Discrete Math.* 1 (1977), 343–362.

[4] Cole, R., Parallel Merge Sort, *SIAM Journal of Computing* 17 (1988), 770–785.

[5] Dekel, E., and Sahni, S., Binary Trees and Parallel Scheduling Algorithms, *IEEE Trans. on Computers* C-32 (1983), 307–315.

[6] Dekel, E., and Sahni, S., Parallel Scheduling Algorithms, *Operations Research* 31 (1983), 24–49.

[7] Dekel, E. and Sahni, S., A Parallel Algorithm for Convex Bipartite Graphs and Applications to Scheduling, *Journal of Parallel and Distributed Computing 1* (1984), 185–205.

[8] Dolev, D., Upfal E. and Warmuth, M., The Parallel Complexity of Scheduling with Precedence Constraints, *Journal of Parallel and Distributed Computing 3* (1986), 553–576.

[9] Frederickson, G. N. Scheduling Unit-Time Tasks With Integer Release Times and Deadlines, *Inf. Proc. Letters* 16 (1983), 171–173.

[10] Frederickson, G. N. Private Communication.

[11] Garey, M. R., Johnson, D. S., Simons, B. B., and Tarjan, R. E., Scheduling Unit-Time Tasks with Arbitrary Release Times and Deadline, *SIAM Journal of Computing* 10 (1981), 256–269.

[12] Helmbold, D. and Mayr, E., Fast Scheduling Algorithms on Parallel Computers, STAN-CS-84-1025, Department of Computer Science, Standford University (1984), 1–27.

[13] Horn, W. A., Some Simple Scheduling Algorithms, *Naval Research Logistics Quarterly* 21 (1974), 177–185.

[14] Jackson, J. K., Scheduling a Production Line to Minimize Tardiness, Management Science Research Project, Univ. California, Los Angelos, CA, Res. Rep. 43, 1955.

[15] Martel, C. U., A Parallel Algorithm for Preemptive Scheduling of Uniform Machines, Technical Report, Computer Science Division, University of California at Davis, (1988), 1–21.

[16] Monma, C. L., Linear-Time Algorithms for Scheduling on Parallel Processors, *Operations Research* 30, (1982), 116–124.

[17] Rodger, S. H., An Optimal Parallel Algorithm for Preemptive Job Scheduling That Minimizes Maximum Lateness, Purdue Computer Science Technical Report CSD-TR-88, (Aug. 1988), 1–31.

[18] Simons, B. B., A Fast Parallel Algorithm for Single Processor Scheduling, *Proc. of the 19th Annual Symp. on Foundations of Computer Science* (1978), 246–252.

Finding a Minimal Cover for Binary Images: an Optimal Parallel Algorithm*

Dipen Moitra
Computer Science Department, Cornell University, Ithaca, NY 14853

Abstract *Given a black and white image, represented by an array of* $\sqrt{n}\times\sqrt{n}$ *binary valued pixels, we wish to cover exactly the black pixels with an irreducible set of (possibly overlapping) maximal squares. We derive an optimal parallel algorithm for this problem, with a linear processor-time product, which for computation times* T *in* [log n, n] *runs on a* CREW-PRAM, and for T in [log n / loglog n, log n) *runs on a* CRCW-PRAM. *The cornerstone of our algorithm is a novel data structure, the* cover graph, *which captures the essential combinatorial covering relationships amongst the maximal squares of the image in a minimized form.*

1 Introduction

In remote sensing it is important to encode digitized images compactly in order to reduce storage and transmission costs, even at the expense of time and hardware for encoding and decoding the raw image. An irredundant encoding of digitized images can also provide the basis for efficient image analysis and manipulation algorithms, for applications such as VLSI mask generation, incremental update of raster displays, and object representation for sequential frames of a dynamic polygonal scene. In the rest of this paper we will discuss parallel algorithms for obtaining a minimal encoding of black and white images. We omit many details and proofs in this abstract, for which the reader is referred to [Moi-88].

A popular technique for image representation is the *Quad-tree* [TaP-75], which provides a recursive decomposition of a binary image into quadrants, each quadrant being decomposed further only if it is neither all white nor all black. This yields a 4-ary tree with at most log n levels. Although this supports efficient algorithms for image manipulation, the size of the resulting tree is extremely sensitive to the placement of the *origin* for the decomposition at the root; for example consider the decomposition of a single black square of side $\sqrt{n}$, starting at (1, 1); it contains $\Omega(\sqrt{n})$ nodes, instead of the single node which suffices for a minimal cover. There have been many attempts at optimal placement of the *origin*, but all are expensive techniques, and do not yield guaranteed size reductions [Sam-84]. Recently Scott and Iyengar [ScI-86] proposed representing binary images using a reduced set of *maximal squares*, but their algorithm does not yield an irreducible cover, is sequential, and requires O(n log n) time. We obtain an optimal parallel algorithm for this problem, which yields an irreducible cover, and has a linear processor-time product for all computation times T in [log n / loglog n, n].

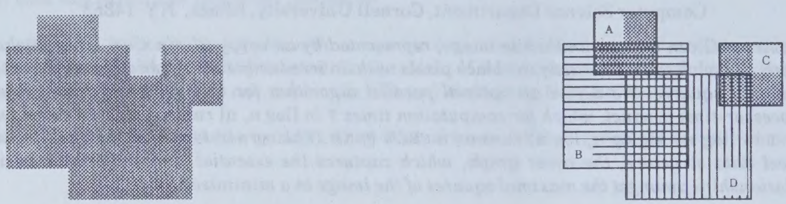
2 Overview of the Minimal Cover algorithm

The input image is given as an array of $\sqrt{n}\times\sqrt{n}$ binary valued pixels. Although the value of a pixel is represented as *true / false*, in the discussion we will use the synonymous values *black / white*. We wish to cover the image with a minimal set of black squares (possibly overlapping). For example, consider the black and white image shown in Figure 1 below, and a minimal cover for it as shown.

---

*Research supported in part by NSF grant DCI-86-02256, ONR contract N00014-83-K-0640, and IBM agreement 12060043.

Definitions A *maximal square* is a black subsquare of the image, not contained in any larger black square. An *essential square* is a maximal square containing at least one pixel not covered by any other maximal square. A *cover* for a given image is a set C of maximal squares whose union equals the image. If no subset of C is a cover, C is called a *minimal cover*.



a. Given black and white image b. Cover with maximal squares

Figure 1. Minimal Cover of Squares

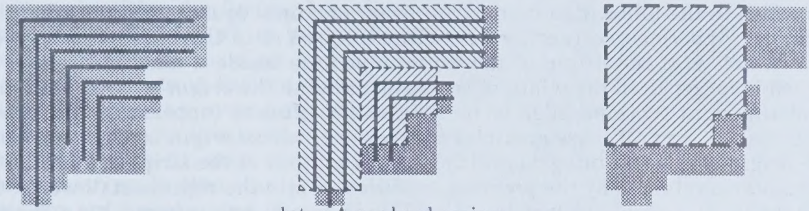
algorithm minimal_cover

1. Determine the maximal squares, using the algorithm of Section 3.
2. Identify rectangular regions, each of which is covered by a contiguous sequence of identically sized maximal squares, and retain a minimum cover for each rectangle, using the algorithm in Section 4.
3. For every retained maximal square, determine if it is covered by the union of its neighboring maximal squares, as in Section 5. If no such cover exists, then mark the square as being *essential*.
4. For every non-essential square m, derive its *extended cover* Ξ_m, which consists of all retained squares which are "useful" for covering m, as given in Section 6.
5. Determine a directed bipartite *cover graph* $G=\langle(M, R), E\rangle$ which embodies all the useful covering relationships within the image, as given in Section 6. M is the set of retained maximal squares, and R consists of the rectangles in a suitable decomposition of the squares in M. The edges in E denote containment, and satisfy the following: (i) a given square is contained in the union of all of its predecessor rectangles; (ii) a given rectangle is contained in every one of its predecessor squares.
6. Determine a minimal cover for the image, using the coloring algorithm of Section 7.

3 Identifying Maximal Squares

Let us define the *largest black square* ${}^{SE}L_{ij}$ to be the largest square of black pixels, with its top left corner (*origin*) at $\langle i, j\rangle$ and its diagonally opposite corner to the *SouthEast* of $\langle i, j\rangle$. We describe below an optimal parallel algorithm, which uses *parallel prefix computations* [KRS-85, CoV-88a], to find ${}^{SE}L_{ij}$ for every pixel $\langle i, j\rangle$. Similarly we obtain ${}^{SW}L_{ij}$, ${}^{NW}L_{ij}$ and ${}^{NE}L_{ij}$. In the sequel, L_{ij} will mean ${}^{SE}L_{ij}$. These initial steps helps us to find the maximal squares efficiently, and also generate information which will be useful in the elimination of redundant maximal squares.

For every black pixel, find R_i, the right end of the largest horizontal black strip containing it; and also L_i, the lower end of the largest vertical black strip containing it. Hence determine the largest L-shaped region which could potentially form the upper-left border of a black square to the right and below: as the minimum of the distances to the two ends. Finally for every pixel, determine the largest square covered by the diagonal sequence of L-shaped black borders to the right and below, thereby determining the largest black square in that direction. (See Figure 2.)



b. two Arrowhead regions

a. gray region, with sequence of L-borders

c. composite Arrowhead region, & square

Figure 2. Optimal algorithm for Finding Largest Black Squares

Given a semigroup $\langle S, \otimes \rangle$ (where $\otimes$ is a binary associative operation on S), a *prefix computation* over a sequence $s_1 \ldots s_n$ of elements of S, consists of computing the n partial products $s_1 \otimes \ldots \otimes s_k$ for all $k \le n$. A *prefix computation* over n elements can be performed optimally in parallel, in time $T \in [\log n, n]$, using an n/T processor CREW-PRAM, and in time $T \in [\log n / \log\log n, \log n)$, using an n/T processor CRCW-PRAM [CoV-88a]. We can transform the problem of finding the longest horizontal / vertical black strips, into a prefix computation as follows. Define the *blackhead* of a subsequence $x_i \ldots x_{i+l_i-1}$ as $B_i \doteq \langle d_i, l_i \rangle$, where d_i (the depth of the *blackhead*) is the maximum number of ones which prefix the subsequence $x_i \ldots x_{i+l_i-1}$. We can multiply the blackheads $\langle d_1, l_1 \rangle$, $\langle d_2, l_2 \rangle$ of two adjacent subsequences $x_i \ldots x_{i+l_1-1}$ and $x_{i+l_1} \ldots x_{i+l_1+l_2-1}$ into a single blackhead $\langle d, (l_1+l_2) \rangle$, where $d \leftarrow$ **if** $d_1 < l_1$ **then** d_1 **else** $d_1 + d_2$ **fi**. Thus the right recursive computation implied by the definition of R_i can be replaced by the associative computation of B_i, and the desired result R_i equals *depth*(B_i).

The last step can be similarly transformed into a prefix computation as follows. Define an *arrowhead* region (i.e. the L-shaped top left border of a square) as $A_{ij} \doteq \langle d_{ij}, e_{ij} \rangle$, where d is the *depth* of its leading edge, and its *extent* e is the thickness of each arm (in pixels). We can multiply two adjacent arrowheads $\langle d1, e1 \rangle$ and $\langle d2, e2 \rangle$, into a single arrowhead $\langle \min(d1, d2+e1), (e1+e2) \rangle$. This operation is also associative, and can be computed along the diagonals of the image, using a parallel prefix computation. When the thickness of an arrowhead equals its depth, we have found a largest black square. L_{ij} equals *depth*(A_{ij}). As Lemma 1 shows, we can now identify the maximal squares with only constant computation per pixel.

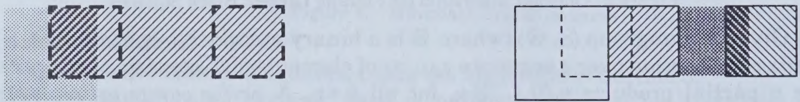
Lemma 1 Let P be the largest black square with its *origin* at $\langle i, j \rangle$, and $l_{ij} \ge 1$ be its width. Then P is not *maximal* iff $l_{i-\delta i,\, j-\delta j} > l_{i\,j}$, for some $\langle \delta i, \delta j \rangle \in \{\langle 0, 1\rangle, \langle 1, 0\rangle, \langle 1, 1\rangle\}$. **proof** (omitted).

4 Dividing Uniform Strips

Having obtained the maximal squares, we identify rectangular regions spanned by sequences of identically sized maximal squares, and retain a minimal cover for each. Without this step, the cover graph can have $\Omega(n \sqrt{n})$ edges, and/or a maximum degree of $\Omega(\sqrt{n})$ in the worst case.

Definition : A horizontal (vertical) *uniform strip* is a maximal sequence of identically sized maximal squares, whose *origins* are horizontally (vertically) contiguous.

We retain a minimum cover for a uniform strip, by dividing its longer side by its width in the following manner. For every maximal square m, determine the leftmost (uppermost) corner (*origin*) of the *uniform strip* U which contains it as shown below. If the *origin* of the uniform strip is inside a maximal square h which is larger than the width of the strip, redefine the *origin* to be shifted just past the right (bottom) edge of h. Retain the leftmost (uppermost) maximal square in U, and all those maximal squares in U whose *origin* is displaced from the origin of U by an integral multiple of the width w of the strip. In addition to the squares retained by the previous condition, retain the rightmost (lowermost) square in U, unless its last $(w-1)$ columns (rows) are covered by a larger maximal square. Discard all the others. (See Figure 3 below.)



a. uniform strip, with black surrounding region

b. squares retained after division

Figure 3. Dividing *Uniform Strips*

In order to find the left corner of (say) a horizontal uniform strip, we must compute the length of the uniform strip to the left of $\langle i, j\rangle$, as $H_{ij} \doteq \min \{k \mid 0<k\leq j$ such that $|M_{i,\,j-k}| \neq |M_{ij}| \}$. This can be transformed to a *prefix computation* by the technique of Section 3. We define a *uniform_tail* U_{ij} to be the right end of a uniform strip which has its *origin* at $\langle i, j-k+1\rangle$, and represent it as the triple $\langle m, d, l\rangle$; where m is the size of the rightmost maximal square in the tail, d (depth) is the length of the uniform strip at the end of the tail, and l is the length of the tail. We can associatively combine two adjacent *uniform_tails* into a single tail, as $\langle m_2, d_2, l_2\rangle \otimes \langle m_1, d_1, l_1\rangle \doteq \langle m_1, d, (l_1+l_2)\rangle$; where $d \leftarrow$ **if** $(m_1 \neq m_2)$ **or** $(d_1 < l_1)$ **then** d_1 **else** $d_1 + d_2$ **fi**. The required computation is defined recursively as $U_{ij} \leftarrow U_{i,j-1} \otimes \langle m_{ij}, m_{ij}, m_{ij}\rangle$. Observe that this computation proceeds from right to left, starting at the *origin* of each maximal square. H_{ij} equals *depth*(U_{ij}).

5 Sufficient conditions for Redundancy

In the worst case it is possible that $\Omega(\sqrt{n})$ squares overlap with a given square m, yet none of them are useful for determining the redundancy of m relative to some selected set of squares. If we restrict our attention to those squares which overlap maximally with m, we can determine efficiently if m is essential, else identify a cover for it, as shown below.

Definition : The *basic cover* β_m for an inessential square m, is a set of maximal squares excluding m, whose union covers m; such that for all $S \in \beta_m$, $S \cap m$ is not strictly contained in $T \cap m$ for any $T \notin \{m, S\}$.

From the definition of a basic cover it follows that a square is not *essential* (i.e. potentially redundant) *iff* it has a basic cover. In the following theorem we prove that if a maximal square m is not essential, its *basic cover* has four or fewer maximal squares.

Theorem 1 The *basic cover* for an inessential square m contains four or fewer squares.

Theorem 2 We can determine whether a given maximal square m is *essential;* and if m is inessential we can identify a *basic cover* for it: all with only a constant number of operations. **proofs** (omitted)

Our algorithm for finding a basic cover relies on examining the length of maximal black strips which border the square m on each side, starting at each corner. This yields the sizes and locations of maximal squares which overlap maximally with m. A little case analysis helps to eliminate those squares which either do not have a cover completion, or whose overlap with m is subsumed by another square. At this point, a given square S which we have found to overlap maximally with m, may already have been discarded during the division of uniform strips. In that case we locate the predecessor / successor P of S in the uniform strip containing S and P. If P overlaps with m, we add P to the *normalized basic cover* of m.

6 Determining the Cover Graph

Now we will determine the geometric covering relationships amongst the maximal squares, and represent these using a *cover graph*. Our goal is to be able to answer efficiently all questions of the form "does this set S of maximal squares cover this square m?"; and also to be able to select an independent set of squares which can be discarded in parallel, while simultaneously guaranteeing that the entire image is covered at all times by the undiscarded squares.

6.1 Deriving the Extended Cover by Refinement

First we need to identify all those squares which can be useful for covering any given square m. If any square P is to be useful, there must be a set S of squares containing P, such that S covers m, but $(S-P)$ does not. Further, we want to avoid choosing squares which cover too small a region of m, since in the worst case we could be attempting to cover each region of m in $\Omega(\sqrt{n})$ different ways, and hence be unable to meet our overall linear resource bounds.

Definition : A pair of maximal squares P and Q are *pairwise essential* if $P \cap Q$ contains a pixel p which does not belong to any other maximal square (other than P or Q). The *extended cover* Ξ_{m} of an inessential square m consists of all those retained squares P, such that (i) P belongs to some irreducible cover of m, and (ii) $(P \cap \mathrm{m})$ is not a strict subset of $(X \cap Y)$ for any squares X and Y which are pairwise essential.

We can determine the extended cover Ξ_{m} of a square m, starting with its *basic cover* β_{m}, and adding to it the basic cover β_S for every square S in β_{m}. Given two squares A, B in Ξ_{m}, we say that $A <_{\mathrm{m}} B$ *iff* $A \cap \mathrm{m}$ is strictly contained in $B \cap \mathrm{m}$. This partial order separates the squares in Ξ_{m} into *chains*, each of which starts with a square that overlaps maximally with m, i.e. is in its basic cover. We show that each chain can be traced in decreasing order of overlap with m, by a constant sequence of basic cover refinements.

Theorem 3 Let P be a maximal square in the extended cover Ξ_{m} of an inessential square m, such that P does not belong to the normalized basic cover β_{m} of m. There exists a square Q in β_{m}, such that P belongs to the basic cover β_Q of Q.
proof (omitted)

6.2 Factoring the Cover Relations in Normal Form

Once we have determined the *extended cover* Ξ_{m} for every square m, we are still faced with the task of efficiently determining if an arbitrary subset of Ξ_{m}

does indeed cover m. We can do this by determining a characteristic function for all irreducible covers of m, which are subsets of Ξ_m. It is obvious that the cover function for a given square m consists of a conjunction of the cover functions for each pixel p in m; and the cover function for each pixel p is a disjunction of the squares which cover p. An obvious reduction is to aggregate the cover functions of all those pixels which are covered by exactly the same set of squares.

Definition The *cover function* for an inessential maximal square m is a boolean formula $F_m(s_1, .., s_k)$, such that (i) A subset S of $\{m_1, .., m_k\}$ is an irreducible cover for m, implies that $F_m(s_1, .., s_k)$ is true; where s_i is the truth value of the predicate $(m_i \in S)$; and (ii) $F_m(s_1, .., s_k)$ is false, implies that S does not cover m.

6.3 Unique Rectangular Decomposition of Squares

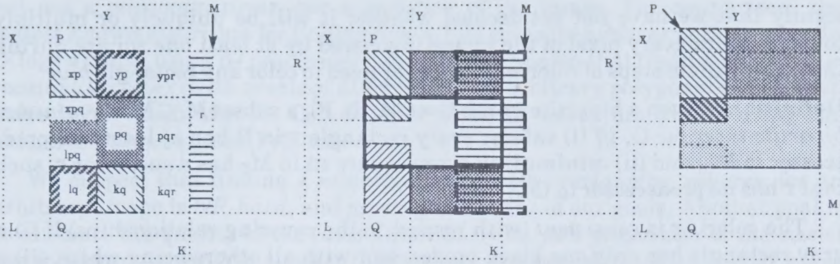
A final reduction of the cover function occurs when a pair of conjuncts of the form $(A \vee B) \wedge (A \vee B \vee C)$ can be simplified to $(A \vee B)$, as shown below. Given a maximal square m, consider the set C_m of all irredundant convex covers of m, given by $C_m = \{x \mid x$ is a set of convex subregions of m whose union covers m; and $(\forall$ distinct $p, q \in x)\ p \not\subseteq q\}$. There is a naturally defined complete partial order $\propto$ on C_m, such that for all pairs x_1, x_2 in C_m, $x_1 \propto x_2$ *iff* $(\forall p \in x_1)(\exists q \in x_2)$ such that $p \subseteq q$. The minimum element $\perp$ of the lattice $\langle C_m, \propto\rangle$ is the pixel level partition of m, and the maximum element $\top$ is $\{m\}$. Without the irredundancy condition in the definition of C_m, we can get two different covers which are less than or equal to each other.

Definitions A *reduced cover* x_m of m is any partition of m into convex regions. Let S be a subset of $(M - \{m\})$ which covers a given maximal square m, and I_S be the set of all subsets of S which constitute an irreducible cover of m. Then a cover $x \in C_m$ is *compatible* with S, if every region r in x can be covered by some single square in any irreducible cover $K \in I_S$. The *induced cover* $\eta_m(S)$ is the least upper bound on the lattice $\langle C_m, \propto\rangle$, of the reduced covers x_m of m which are compatible with S.

Theorem 4 Let S be a set of maximal squares whose union contains a given maximal square m, and $m \notin S$. Then the *induced cover* $\eta_m(S)$ is *compatible* with S.

proof Partition m into maximal connected regions, such that all the pixels in a given region are contained in the same subset of S. Label each region with the largest subset of S which contains it. Mark those regions r as *residual*, for which no other region t in the partition of m is labelled with a proper subset of the squares which label r. For every non-residual region t, add it to every residual region r which is labelled with a proper superset of the squares which label t. Finally discard all the non-residual regions. Call the remaining set of augmented (residual) regions as $\eta_m(S)$. It is obvious that $\eta_m(S)$ as constructed is compatible with S. A little thought will convince the reader that $\eta_m(S)$ as constructed is also the *lub* of the covers of m compatible with S. (See Figure 4) □

The cover function F_m also has an obvious geometric interpretation. Each conjunctive term in F_m is a rectangle which is the intersection of the squares which are disjuncts in that term, and m equals the union of all such rectangles defined by the disjuncts in F_m. We can easily represent the cover function F_m for a square m, via a bipartite graph $G = (\langle M, R\rangle, E)$ which is isomorphic to F_m; M is



compatible partition of M, residual regions are shaded

induced cover for M $M \subseteq (X \vee P)(Y \vee P)(P \vee Q)(L \vee Q)(K \vee Q)R$

induced cover for P $P \subseteq (Q \vee M) \wedge X \wedge Y$

Figure 4. Derivation of Induced Covers for a collection of squares

the set of squares ($m \cup \Xi_m$), R is the set of rectangles in the cover of m induced by Ξ_m, and the edges in E denote containment relationships amongst squares and rectangles. We can obtain the cover graph for the entire image by unioning the cover graphs for each of the undiscarded squares. During this step, we also apply the graph transformation shown in Figure 5, to eliminate useless conjuncts, to get the *reduced cover graph* shown in Figure 6.

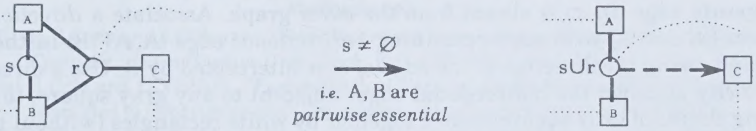
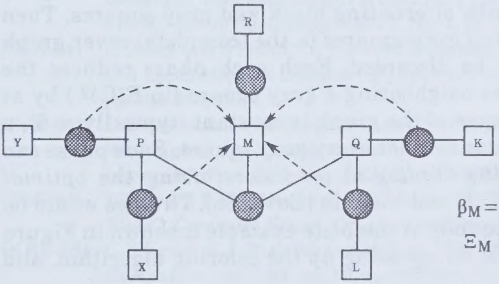


Figure 5. Deleting Useless Conjuncts



$\beta_M = \{P, Q, R\}$; $\beta_P = \{X, Y, M\}$; $\beta_Q = \{K, L, M\}$;

$\Xi_M = \beta_M \cup \beta_P \cup \beta_Q = \{P, Q, R, X, Y, K, L\}$

Figure 6. Reduced Cover-graph for the collection of squares in fig. 4

Theorem 5 The *reduced cover graph* has a maximum degree bounded by a (low) constant. **proof** (omitted)

7 Finding a Minimal Cover

We find a minimal cover for the image by determining a 2-coloring of the *squares* in the cover graph G, using *white* (discard) and *black* (retain). The *rectangles* are also colored in the manner implied by the containment relationships in G. We need three colors to represent the covering state of a rectangle: *black* (multiply covered), *gray* (uniquely covered) and *blue* (not covered). Before starting execution of the graph coloring algorithm we color all the squares *gray*, signifying that we are yet undecided whether to retain or discard them. We also color all the rectangles *white*, and let a *white rectangle* signify that we have not yet decided whether it will be uniquely or multiply covered. Since every pixel of the image is covered by at least one square during the intermediate steps of coloring, we never need to color any rectangle *blue*.

Definition : Given a bipartite graph $G=(\langle M, R\rangle, E)$, a subset $M_C \subseteq M$ is a *minimal bipartite cover* for G, *iff* (i) **safety**: every rectangle r in R has at least one predecessor in M_C, and (ii) **minimality**: every square m in M_C has a successor r, such that r has no predecessor in $(M_C-\{m\})$.

The coloring is *consistent* (with respect to the covering relationships) if (i) a gray rectangle has only one black predecessor with all others being white, (ii) a black rectangle has two or more black predecessors, and (iii) a white rectangle has no more than one black predecessor. During the execution of the graph coloring algorithm we maintain *consistency,* and also the following invariants: (i) **safety**: Every rectangle has at least one black or gray predecessor. (ii) **minimality**: A *black* square has at least one gray or white successor. The algorithm terminates when no square remains *gray*, and no rectangle remains *white*. The conjunction of the termination predicate with the invariants implies that a *minimal cover* has been found.

Start the coloring algorithm by suppressing all edges $\langle r, m\rangle$ for which the opposite edge $\langle m, r\rangle$ is absent from the cover graph. Associate a *direction label* from {N, ... NW} with each remaining *bidirectional* edge $\langle A, A\cap B\rangle$ in the cover graph, using the direction of the edge(s) of A intercepted by B. Use a directional priority amongst the bidirectional edges adjacent to any gray square, to obtain long chains of gray squares joined together by white rectangles (without tokens) and bidirectional edges. Suppress the rectangles, and derive a chain of gray squares. Color each such chain with alternating black and gray squares. Then consider the status of the remaining gray squares in the (complete) cover graph to determine if any of them can be discarded. Each such *phase* reduces the maximum number of gray squares neighboring a gray square (in $E(G)^2$) by at least two. Since the maximum degree of the graph is constant (typically ≤ 6), a constant number of *phases* suffices to re-color every gray square. Each phase can be executed in O(*log* n) time using O(n/*log* n) processors, using the *optimal parallel list ranking* technique of Cole and Vishkin [CoV-88b]. Thus we would be able to meet our overall resource bounds. A complete example is shown in Figure 7. Many optimizations are possible for speeding up the coloring algorithm, and are discussed in the full paper.

8 Conclusions

Ferrari et. al. [FSS-84] give a sequential algorithm for finding the *minimum rectangular partition* of a binary image, but their algorithm relies on finding a minimum vertex cover for a bipartite graph. There exist polygons with a rectangular cover of k rectangles, that need $\Omega(k^2)$ rectangles for a rectangular partition. Also there does not seem to be any simple way of deriving a deterministic parallel (NC) algorithm utilizing their results since, for an image with holes, their cover graph is not convex. It is known that finding the minimum vertex cover for an arbitrary graph is *NP-hard*, although the problem is in NC for *convex bipartite* graphs [DeS-84].

By using overlapping *maximal squares* for covering a binary image, we were able to determine an *irreducible cover* in O(log n) time using n / log n processors, but not a *minimum* cover, nor a *partition* of the image. For comparison, the fastest known algorithm for finding a minimal cover for arbitrary graphs runs in $O(\log^2 n)$ time, using n^3 processors [Lub-85]. It is known that finding a *minimum rectangular cover* (with overlaps allowed), for an arbitrary polygonal region with holes is *NP-hard* [Mas-79, CoR-87]. For polygons which are convex along one axis, $O(n^2)$ time suffices [FrK-84].

We suspect that finding a *minimum cover* with *overlapping squares*, for an arbitrary image is *NP-hard*, and proving this is one of our goals. Another goal is to estimate the quality of the cover obtained by us, as a proportion of the size of the minimum cover. We suspect that we can produce covers that are no worse than 7/6 times the minimum in the worst case, and are considering heuristics which could bound this even better. We are also exploring the possibility of embedding the cover we obtain into a search tree, in order to obtain a reduced image representation which could compete with *Quad-trees* [Sam-84] for such operations as finding the union and intersection of images.

Acknowledgements I gratefully acknowledge the many constructive discussions with Gianfranco Bilardi, Kieran Herley, S. Seetharam Iyengar, Kevin Karplus, Abha Moitra and Joseph Mitchell.

References

[CoR-87] H.E.Conn and J.O'Rourke, "Some Restricted Rectangle Covering Problems", *Dept. of Comp. Sc., Johns Hopkins Univ.*, Tech. Rep. JHU-87/13, June 1987.

[CoV-88a] R.Cole and U.Vishkin, "Faster Optimal Parallel Prefix Sums and List Ranking", submitted to *Information and Computation.* (also Courant Institute CS-TR-277)

[CoV-88b] R.Cole and U.Vishkin, "Approximate Parallel Scheduling, Part I: the basic technique, with applications to Optimal Parallel List Ranking in Logarithmic Time", *SIAM Jour. of Computing*, vol. 17, No. 1, pp. 128-142, Feb. 88.

[DeS-84] E.Dekel and S.Sahni, "A Parallel Matching Algorithm for Convex Bipartite Graphs and Applications to Scheduling", *Jour. of Parallel and Distributed Computing*, vol. 1, pp. 185-205, 1984.

[FrK-84] D.S.Franzblau and D.J.Kleitman, "An Algorithm for Constructing Regions with Rectangles: Independence and Minimum Generating Sets for Collections of Intervals", *Proc. 16th Ann. ACM Symp. on Theory of Computing*, pp. 167-174, 1984.

[FSS-84] L.Ferrari, P.V.Sankar and J.Sklansky, "Minimal Rectangular Partitions of Digitized Blobs", *Computer Vision, Graphics and Image Processing*, vol. 28, pp. 58-71, Oct. 84.

[KRS-85] C.P.Kruskal, L.Rudolph and M.Snir, "The Power of Parallel Prefix", *IEEE Trans. on Computers*, vol. C-34, no. 10, pp. 965-968, Oct. 1985.

[Lub-85] M.Luby, "A Simple Parallel Algorithm for the Maximal Independent Set Problem", *Proc. 17th Ann. ACM Symp. on Theory of Computing*, pp. 1-10, 1985.

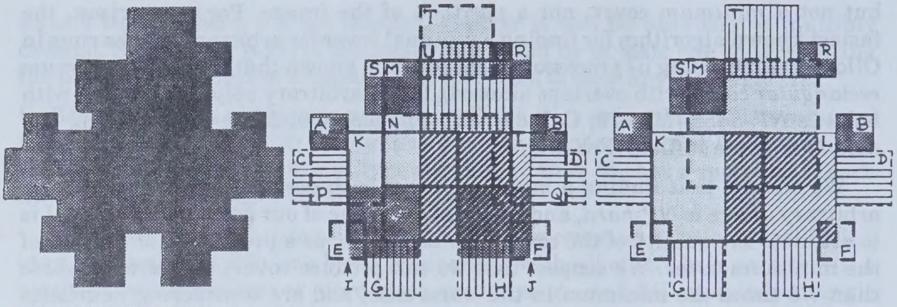
[Mas-79] W. J. Masek, "Some NP-complete Set Covering Problems", unpublished manuscript MIT, 1979.

[Moi-88] D. Moitra, "Finding a Minimal Cover for Binary Images: an Optimal Parallel Algorithm", *Cornell University*, Tech. Rept. in preparation.

[Sam-84] H. Samet, "The Quadtree and Related Hierarchical Data Structures", *ACM Computing Surveys*, vol. 16, pp. 187-260, 1984.

[ScI-86] D.S.Scott and S.S.Iyengar, "TID : a Translation Invariant Data Structure for Storing Images", *Comm. of the ACM*, vol. 29, no. 5, pp. 418-429, May 1986.

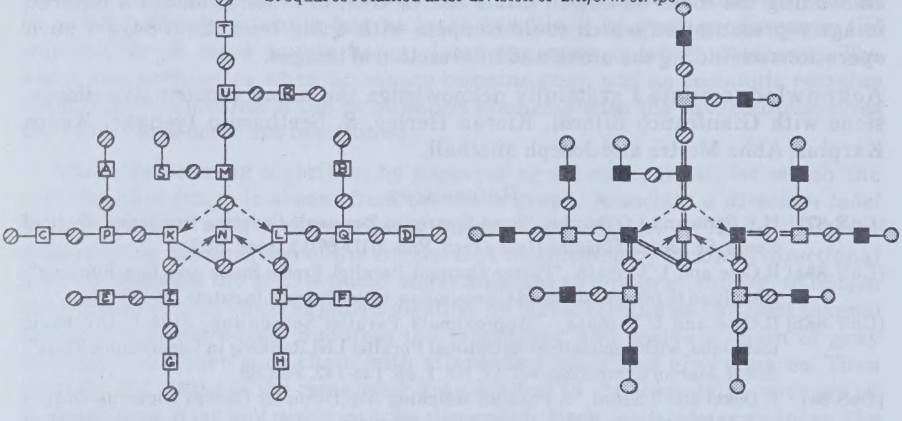
[TaP-75] S. L. Tanimoto and T. Pavlidis, "A Hierarchical Data Structure for Picture Processing", *Computer Graphics and Image Processing*, vol. 4, pp. 104-119, 1975.


T
U
R
SM
N
A
K
B
L
C
D
P
Q
E
F
I
G
H
J

a. given black & white image

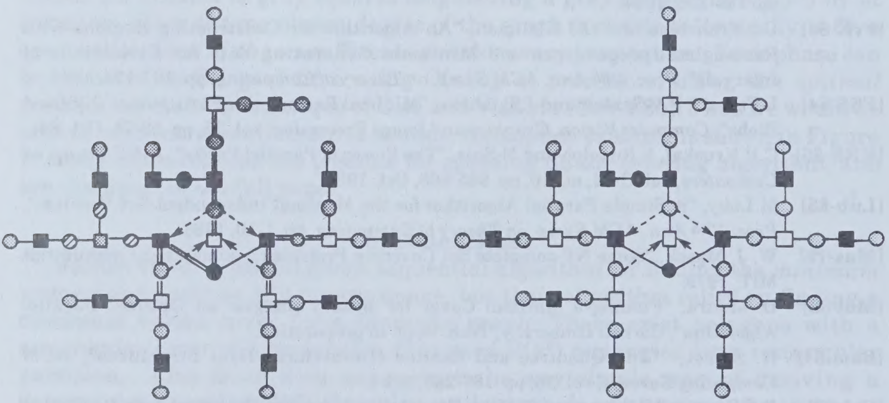
g. irreducible cover obtained

b. squares retained after division of *uniform strips*


T
U
R
A
S
M
B
K
N
L
C
P
Q
D
E
I
J
F
G
H

c. reduced cover graph for the image

d. after marking *essential* squares, and coloring chains *black-gray*



e. after stitching extracted chains back into the graph

f. coloring state after applying *black rule*

Figure 7. A complete example for the minimal cover algorithm

STABILITY ANALYSIS OF ALMOST SINUSOIDAL PERIODIC OSCILLATIONS IN NONLINEAR CONTROL SYSTEMS SUBJECTED TO NONCONSTANT PERIODIC INPUT

G.S. KRENZ
Dept. of Math., Stat. and Computer Sci.
Marquette University
Milwaukee, WI 53233

R.K. MILLER
Department of Mathematics
Iowa State University
Ames, IA 50010

ABSTRACT.

We investigate the existence, local uniqueness and local stability properties of almost sinusoidal periodic oscillations in a class of nonlinear control systems subjected to a nonconstant periodic input.

Provided two parameters are sufficiently small, a modified Routh-Hurwitz condition is given which determines the stability of the forced response. The analysis uses the classical single-input sinusoidal describing function to predict the amplitude and phase shift of the fundamental component of the forced response; a novel linearization of the forced problem; averaging; and a simple theorem concerning perturbed linear systems.

We present several systems which, in theory, satisfy our results. A specific example demonstrates how the results could be used in practice.

1. INTRODUCTION.

In this paper, we investigate the stability of periodic motions in a class of nonlinear control systems subjected to continuous, nonconstant, periodic inputs. In particular, we use the single-input sinusoidal describing function method [4] to obtain the approximate amplitude, a_1, and phase shift, α, of the system response. We employ several state-space coordinate transformations, averaging and a result on perturbed linear systems to

1. verify the existence and uniqueness of a periodic motion $x_p(t)$ near the approximate solution determined by a_1 and α, and
2. analyze the stability properties of $x_p(t)$.

The class of control systems consists of a linear part and a nonlinear part connected in a single loop feedback configuration (see Figure 1). The linear part is given by a controllable and observable realization [7] of a real rational transfer function $G(s)$, where the degree of the numerator is less than the degree of the denominator of $G(s)$. The nonlinear part of the system is required to be an odd, continuous, single-valued function with some additional piecewise differentability properties.

The most popular method of testing stability of periodic solutions of unforced nonlinear feedback systems is the quasistatic stability analysis (also called the Loeb criterion). This analysis is usually done graphically. Our results can be viewed as an extension of the quasistatic stability analysis to forced nonlinear feedback systems. Our result does not have a graphical interpretation. Instead, stability is checked in the Routh-Hurwitz style by determining the signs of certain easily computed parameters. Stability results are derived and proved by techniques similar to those in the unforced case [11,12]. However, the presence of the forcing term causes sufficient complications so that the results given here do not constitute mere modifications or obvious extensions of earlier results.

There is extensive literature devoted to the theoretical justification of the describing function method as it is currently used in studying limit cycle behavior in nonlinear systems. The results of Bass [1], Bergen and Franks [2], Bergen et al. [3], Mees and Bergen [8], Skar et al. [14], and Swern [15] are concerned with the existence of self-sustained oscillations in systems subjected to zero inputs. On the other hand, Holtzman [5], Miller and Michel [9], and Sandberg [13] used the describing function method to obtain sufficient conditions which guarantee the existence of periodic solutions of nonlinear control systems subjected to periodic inputs.

2. STATEMENT OF MAIN RESULT.

We will analyze feedback systems of the form displayed in Figure 1, where for some $\omega > 0$, $r_0 \in R$ is continuous and $2\pi/\omega$-periodic. Without loss of generality, we may assume r_0 has the form

$$r_0(t) = a_0 \sin \omega t + \psi(\omega t),$$

where

$$a_0 > 0 \text{ and } \int_0^{2\pi} \psi(t) e^{it}\, dt = 0.$$

The linear part of the feedback system is denoted by the transfer function $G(s)$. We assume that $G(s)$ is a real rational function, i.e.,

$$G(s) = p(s)/q(s),$$

where,

$$\begin{aligned} p(s) &= \gamma_{J-1} s^{J-1} + \gamma_{J-2} s^{J-2} + \cdots + \gamma_1 s + \gamma_0 \text{ and} \\ q(s) &= s^J + \delta_{J-1} s^{J-1} + \cdots + \delta_1 s + \delta_0, \quad \delta_k, \gamma_k \in R,\ 0 \le k < J. \end{aligned}$$

Some of the leading coefficients of $p(s)$ may be zero, i.e., $0 \le \deg p(s) < J$. In addition, we assume $p(s)$ and $q(s)$ have no common roots. The nonlinear part, $n(\cdot)$, must be an odd function which satisfies some additional smoothness requirements (see (H-2) of Theorem 1.)

Applying the describing function method [4] to the system in Figure 1, we obtain

$$\left(1 + G(i\omega) N(a)\right) a e^{-i\alpha_0} = G(i\omega) a_0, \tag{1}$$

where $N(a)$ is the sinusoidal-input describing function for the nonlinear function $n(y)$. The term $\exp(-i\alpha_0)$ corresponds to the phase shift required to balance the resulting fundamental components of the signals in the system.

Suppose there is a value $a_1 > 0$ for which (1) holds, that is,

$$\begin{aligned} |1 + G(i\omega) N(a_1)| a_1 &= |G(i\omega)| a_0, \text{ and} \\ \alpha_0 &= \arg\left(G(i\omega)^{-1} + N(a_1)\right). \end{aligned} \tag{2}$$

We assume a_1 is a value for which $n'(a_1)$ exists. For example, if n is the saturation function

$$n(y) = \begin{cases} my & |y| \le \delta, \\ m\delta \operatorname{sgn} y & |y| > \delta, \quad m, \delta > 0 \text{ and } \operatorname{sgn} y = y/|y|, \end{cases}$$

then $a_1 \ne \delta$.

The assumption that $p(s)$ and $q(s)$ have no common roots implies the feedback system in Figure 1 has a natural controllable, observable, phase space realization

$$x_0' = A x_0 + b_0 u, \tag{3}$$

$$y_0 = {h_0}^T x_0, \tag{4}$$

with transfer function $G(s) = \mathcal{L}\left\{{h_0}^T e^{At} b_0\right\}$ [7]. Here, A is the companion matrix for $q(s)$,

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -\delta_0 & -\delta_1 & -\delta_2 & \cdots & -\delta_{J-1} \end{bmatrix},$$

while b_0, h_0 are real J-tuples with ${b_0}^T = (0,0,\ldots,0,1)$ and ${h_0}^T = (\gamma_0, \gamma_1, \ldots, \gamma_{J-1})$. The control, u, will be $r_0(t) - n(y_0(t))$.

If we substitute the control $\tilde{u}(t) = a_0 \sin \omega t - N(a_1) y_0(t)$ for u in (3)–(4), we see that the new system has a periodic solution

$$\tilde{x}_0(t) = a_1 E^{-1} (\sin(\omega t - \alpha), \omega \cos(\omega t - \alpha), -\omega^2 \sin(\omega t - \alpha), \ldots)^T,$$

where $\alpha = \alpha_0 + \arg(p(i\omega))$ and $E = |p(i\omega)|$. We will show that there exists a $2\pi/\omega$-periodic solution $x_p(t)$ of the original system (3)–(4) near $\tilde{x}_0(t)$, provided two computable parameters are small. In addition, we will give a stability criterion for $x_p(t)$. Here, stability will mean local asymptotic stability, while instability will be in the sense of Lyapunov [10]. In order to state the stability result for the above phase space realization of the feedback system, we introduce the following notation:

Let $d(s)$ be given by

$$\begin{aligned} d(s) &= q(s) + p(s)N(a_1) - \frac{a_0 E}{a_1}\{(\sin\alpha)\frac{s}{\omega} + \cos\alpha\} \\ &= d_0 s^J + d_1 s^{J-1} + \cdots + d_{J-1}s + d_J, \end{aligned} \qquad \begin{matrix}(5)\\(6)\end{matrix}$$

where α_0 and a_1 are given by equation (2). Define $\hat{\beta}_1$ and $\hat{\beta}_2$ by

$$\hat{\beta}_1 - i\hat{\beta}_2 = \frac{2}{\omega d'(i\omega)}.$$

For $k = 1, 2, \ldots, J-2$, we define Hurwitz determinants D_k associated with $d(s)$ by

$$D_1 = d_1, \quad D_2 = \det\begin{bmatrix} d_1 & d_3 \\ d_0 & d_2 \end{bmatrix}, \quad D_3 = \det\begin{bmatrix} d_1 & d_3 & d_5 \\ d_0 & d_2 & d_4 \\ 0 & d_1 & d_3 \end{bmatrix},$$

$$D_4 = \det\begin{bmatrix} d_1 & d_3 & d_5 & d_7 \\ d_0 & d_2 & d_4 & d_6 \\ 0 & d_1 & d_3 & d_5 \\ 0 & d_0 & d_2 & d_4 \end{bmatrix},$$

and so forth. We take $d_j = 0$ if $j > J$. Define D_{J-1} and D_J by

$$D_{J-1} = 1 + a_1 N'(a_1) \mathrm{Re}\left\{\frac{G(i\omega)}{1 + G(i\omega)N(a_1)}\right\}$$

and

$$D_J = \mathrm{Re}\left\{(\hat{\beta}_1 - i\hat{\beta}_2)(2[q(i\omega) + p(i\omega)N(a_1)] + a_1 N'(a_1) p(i\omega))\right\}.$$

We now present our main result.

Theorem 1. Suppose that system (3)–(4) satisfies

(H-1) all d_j are real for $0 \le j \le J$, with $d_0 = 1$ and that $D_j \ne 0$, for $1 \le j \le J$,

(H-2) the function n is an odd, continuous, piecewise continuously differentiable function. Moreover, on any interval $y_a < y < y_b$, where $n'(y)$ exists and is continuous, $n''(y)$ also exists and is uniformly continuous. The describing function for $n(y)$ will be denoted by $N(a)$,

(H-3) the polynomials $p(s)$ and $q(s)$ have no common factors, and

(H-4) there exists $a_1 > 0$ satisfying (2) and an associated α such that $d(s)$ has the simple roots $\pm i\omega$. In addition, a_1 must be a point of continuity of $n'(y)$.

If $\hat{\beta}_1$ and $\hat{\beta}_2$ are sufficiently small, then there is a (sup norm) neighborhood N_0 of $(t, \tilde{x}_0(t))$ with the properties:

(i) There exists a $2\pi/\omega$-periodic solution $x_p(t)$ of (3)–(4) such that $(t, x_p(t)) \in N_0$ for all $t \in R$.

(ii) If $(t_0, \eta) \in N_0$, but $\eta \neq x_p(t_0)$, then the solution with initial condition $x_0(t_0) = \eta$ must leave N_0 in finite time. Hence, $x_p(t)$ is the only $2\pi/\omega$-periodic solution of (3)–(4) near $\tilde{x}_0(t)$.

(iii) If $D_j > 0$, for $j = 1, 2, \ldots, J$, then $x_p(t)$ is stable.

(iv) If $D_k < 0$, for some k, $1 \leq k \leq J$, then $x_p(t)$ is unstable. □

Since $|x_p(t) - \tilde{x}_0(t)|$ is generally small, we have

$$\begin{aligned}
y_p(t) &= h_0^T x_p(t) \\
&\approx \frac{a_1}{E}\left[\gamma_0 \sin(\omega t - \alpha) + \gamma_1 \omega \cos(\omega t - \alpha) + \cdots + \gamma_{J-1} \frac{d^{J-1}}{dt^{J-1}} \sin(\omega t - \alpha)\right] \\
&= \frac{a_1}{E} \mathrm{Im}\left[p(i\omega) e^{i(\omega t - \alpha)}\right] \\
&= a_1 \mathrm{Im}[\exp(i(\omega t - \alpha + \arg p(i\omega)))] \\
&= a_1 \sin(\omega t - \alpha_0).
\end{aligned}$$

Finally, suppose we have a controllable and observable system

$$x' = A_1 x + b_1 u, \tag{7}$$
$$y = c^T x, \tag{8}$$

with transfer function $G(s) = p(s)/q(s)$. Here, $A_1 \in R^{J \times J}$, $b_1, c \in R^J$ and $u \in R$, with $u = r_0(t) - n(y)$. Then, by results from control theory [7], there is a nonsingular change of coordinates, $x = P x_0$, which brings the system (7)–(8) to the form (3)–(4).

3. EXAMPLES.

Let

$\omega = 1$;

$\alpha_0 = \frac{\pi}{4}$;

$a_1 > 0$ be a fixed, but arbitrary, real number;

$\psi(t) \equiv 0$;

$n(y)$ be any nonlinear function satisfying (H-2) and (H-3) of Theorem 1, such that its describing function, $N(a)$, satisfies the property $N'(a_1) > 0$;

$p(s) = s$; and

$q(s) = s^4 + (k+1)s^3 + (k+1)s^2 + (k + 2 - N(a_1))s + k - 1$, where k is a parameter, with $k > 1$.

In addition, assume $n'(a_1)$ exists.

As required for controllability and observability, $p(s)$ and $q(s)$ do not have a common root. Moreover, evaluation at $s = i$ yields

$$[1 + G(i\omega)N(a_1)]\, a_1 \exp(-i\alpha_0) = G(i\omega) a_1 \sqrt{2}.$$

Thus, for $a_0 = a_1\sqrt{2}$, we have a solution to the describing function equation (1).

Using the above parameters, we obtain

$$\alpha = 3\pi/4,$$

$$\hat{\beta}_1 - i\hat{\beta}_2 = \frac{-(k+1) - i(k-1)}{2(k^2+1)},$$

$$\epsilon = (\hat{\beta}_1^2 + \hat{\beta}_2)^{1/2} = \left[2(k^2+1)\right]^{-1/2},$$

$$d(s) = (s^2+1)(s+1)(s+k),$$

$$D_3 = 1 + a_1 N'(a_1)/2 > 0, \text{ and}$$

$$D_4 = \frac{4k + (k-1)a_1 N'(a_1)}{2(k^2+1)} > 0.$$

Since the roots of $d(s)$ are $\pm i$, -1 and $-k$, we see that it is unnecessary to check D_1 and D_2. For all sufficiently large k, we may apply Theorem 1 to obtain an asymptotically stable periodic solution [6, pp. 31–36].

In the above example, admissible choices of $n(y)$ include:

1. $n(y) = y^p$, where p is any odd integer greater than 1,
2. $n(y) = |y|y^p$, where p is any odd positive integer, or
3. any threshold nonlinearity, with $a_1 > \delta$ (see Figure 2).

Furthermore, the condition $N'(a_1) > 0$ is not essential in the above example. In particular, if $n(y)$ is an ideal saturation function (see Figure 3) and $a_1 > \delta$, then

$$a_1 N'(a_1) = \frac{-4m\delta}{\pi a_1}\sqrt{1 - \left(\frac{\delta}{a_1}\right)^2}.$$

Thus, provided $N'(a_1) \neq 0$ and $D_3 > 0$, similar computations hold without the requirement $N'(a_1) > 0$. That is, for an ideal saturation nonlinearity (by taking a_1 and k sufficiently large) the above choice of ω, α_0, $p(s)$ and $q(s)$ yields an asymptotically stable periodic solution.

Although we have presented an existence, uniqueness and exact stability analysis based upon the describing function method, we concede that the task of checking the "sufficiently small" hypothesis is formidable. Thus, in actual practice, we urge control engineers to use the above theory as they currently use the describing function method:

> when the theory predicts that the system has the desired response properties, verify by simulation.

The following example illustrates this approach. Let

$$G(s) = \frac{s+1}{s(s+2)(s+3)}, \text{ and } n(y) = \begin{cases} 4y & \text{for } |y| \leq 1, \\ 4|y|/y & \text{for } |y| > 1. \end{cases}$$

The function $n(y)$ is an ideal saturation function (see Figure 3, with $m = 4$, $\delta = 1$). Observe that the describing function equation (1) is satisfied when $a_0 \approx 257.6772484$, $a_1 \approx 2.4754145$ ($N(a_1) = 2$), $\omega = 10$ and $\alpha_0 \approx 2.745118723$ (radians). In addition, we have $d(s) = (s^2 + 100)(s+5)$, $\alpha \approx -2.06693891 + 2k\pi$, k any integer, $\hat{\beta}_1 - i\hat{\beta}_2 = (-2 - 1i)/2500$, $D_1 > 0$, $D_2 > 0$ and $D_3 > 0$.

Since $\hat{\beta}_1$ and $\hat{\beta}_2$ are "small", we expect that the system (3)–(4) has a locally asymptotically stable periodic solution near

$$\tilde{x}_0(t) = \frac{a_1}{\sqrt{101}}(\sin(10t - \alpha), 10\cos(10t - \alpha), -100\sin(10t - \alpha))^T.$$

We conducted various simulations using initial conditions near $\tilde{x}_0(t)$. The numerical evidence supports the existence of a locally asymptotically stable periodic solution near $\tilde{x}_0(t)$ [6, pp. 37–38, 42–47].

4. CONCLUDING REMARKS.

We have presented an exact stability analysis for nonlinear systems subjected to continuous, nonconstant, periodic inputs. More precisely, if a system of the form (3)–(4) satisfies hypothesis (H-1) through (H-4), and provided

$$\epsilon = |\hat{\beta}_1 - i\hat{\beta}_2| = \left|\frac{2}{\omega d'(i\omega)}\right|$$

is sufficiently small, then

1. the existence of a $2\pi/\omega$ periodic state-space solution is guaranteed,
2. the $2\pi/\omega$ periodic solution is unique in the sense that it it is the only solution which remains (for all $t \in R$) in a particular neighborhood of an "approximate solution" predicted by the describing function technique, and
3. the local stability (asymptotic stability or instability) of the state-space solution is easily obtained from the linearization of the problem. The stability of the system is checked by a modified Routh-Hurwitz criterion.

Although we have used a describing function approximation, our results are for an actual periodic solution of the nonlinear system. That is, we have analyzed the actual system and its actual response, not an approximate system or an approximate system response.

REFERENCES

[1] R. W. Bass. Mathematical legitimacy of equivalent linearization by describing functions. In *Automatic and Remote Control. Proc. 1st Int. Congr. Int. Fed. Automat. Control*, vol. 2. Ed. J. F. Coales. London: Butterworths, 1961.

[2] A. R. Bergen and R. L. Franks. Justification of the describing function method. *SIAM J. Control Optimization*, **9**, No. 4 (Nov. 1971), 568–589.

[3] A. R. Bergen, L. O. Chua, A. I. Mees and E. W. Szeto. Error bounds for general describing function problems. *IEEE Trans. Circuits & Syst.*, **CAS-29**, No. 6 (June 1982), 345–354.

[4] A. Gelb and W. E. Vander Velde. *Multiple-Input Describing Functions and Nonlinear System Design*. New York: McGraw-Hill, Inc., 1968.

[5] J. M. Holtzman. Contraction maps and equaivalent linearization. *Bell Syst. Tech. J.*, **46**, No. 10 (Dec. 1967), 2405–2435.

[6] G. S. Krenz and R. K. Miller. Stability analysis of almost sinusoidal periodic oscillations in nonlinear control systems subjected to nonconstant periodic input. Milwaukee: Marquette University, Department of Mathematics, Statistics and Computer Science Technical Report No. 288, 1988.

[7] E. B. Lee and L. Markus. *Foundations of Optimal Control Theory*. New York: John Wiley & Sons, Inc., 1967.

[8] A. I. Mees and A. R. Bergen. Describing functions revisited. *IEEE Trans. Autom. Control*, **AC-20**, No. 4 (Aug. 1975), 473–478.

[9] R. K. Miller and A. N. Michel. On existence of periodic motions in nonlinear control systems with periodic inputs. *SIAM J. Control Optimization*, **18**, No. 5 (Sept. 1980), 585–598.

[10] R. K. Miller and A. N. Michel. *Ordinary Differential Equations*. New York: Academic Press, Inc., 1982.

[11] R. K. Miller, A. N. Michel and G. S. Krenz. On the stability of limit cycles in nonlinear feedback systems: analysis using describing functions. *IEEE Trans. Circuits & Syst.*, **CAS-30**, No. 9 (Sept. 1983), 684–696.

[12] R. K. Miller, A. N. Michel and G. S. Krenz. Stability analysis of limit cycles in nonlinear feedback systems using describing functions: improved results. *IEEE Trans. Circuits & Syst.*, **CAS-31**, No. 6 (June 1984), 561–567.

[13] I. W. Sandberg. On the response of nonlinear control systems to periodic input signals. *Bell Syst. Tech. J.*, **43**, No. 3 (May 1964), 911–926.

[14] S. J. Skar, R. K. Miller and A. N. Michel. On existence and nonexistence of limit cycles in interconnected systems. *IEEE Trans. Autom. Control*, **AC-26**, No. 5 (Oct. 1981), 1153–1169.

[15] F. L. Swern. Analysis of oscillations in systems with polynomial-type nonlinearities using describing functions. *IEEE Trans. Autom. Control*, **AC-28**, No. 1 (Jan. 1983), 31–41.

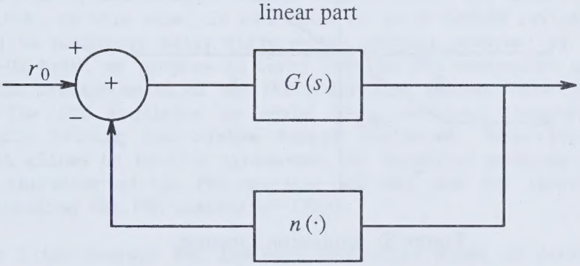


Figure 1. Block diagram of the system.

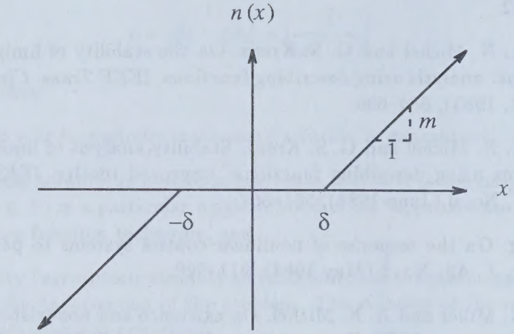


Figure 2. Threshold nonlinearity.

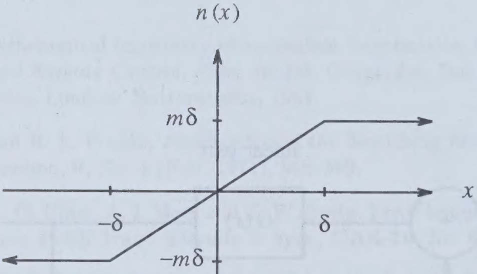


Figure 3. Saturation function

$$n(x) = \begin{cases} mx, & \text{for } |x| \le \delta, \\ m\delta |x|/x, & \text{for } |x| > \delta. \end{cases}$$

PULSE-WIDTH-MODULATION CONTROLLER DESIGN FOR NONLINEAR DELAY DIFFERENTIAL SYSTEMS

HEBERTT SIRA-RAMIREZ
Control Systems Department Universidad de Los Andes Mérida- VENEZUELA

ABSTRACT

In this article a design method is proposed for the specification of Pulse-Width-Modulation (PWM) feedback controllers of Nonlinear Delay Systems. The design method is based on an average model of the PWM controlled system, derived by an infinte sampling frequency assumption which captures the essential qualitative stability properties of the actual controlled system. Some examples are provided.

1. INTRODUCTION

Early contributions to the study of Pulse-width-modulation (PWM) controllers are those of Tsypkin [1]. Further developments were contributed, later on, by Skoog and Blankenship [2], La Cava *et al.* [3] and many others. In all these works, emphasis was primarily placed on the discrete-time aspects of such controllers.

A different design approach [4],[5] has been proposed which uses the geometrical properties of average PWM controlled responses (obtained by an infinite frequency sampling assumption). The results, aside from allowing exact analysis of average responses, found an ideal equivalence with Variable Structure controlled trajectories undergoing sliding motions on the integral manifold of the average PWM system. The class of PWM systems treated in [4]-[5] corresponded to single ON-OFF controlled switch (the control variable takes values in the discrete set {0,1}).

A related class of switch controlled systems is represented by those in which the control variable takes values on the set {-1,0,+1} (See [2]). Typically, gas reaction jets controlling reorientation, or detumbling, maneuvers in artificial satellites are expressible as systems of this class. Some torque actuators, used for control of joint poisitions in robotic manipulators, are also of this type. The controlled switch, in this case, is addressed as an ON-OFF-ON switch. Extending the results of [5] to nonlinear delay differential systems governed by PWM controllers of the ON-OFF-ON type, we propose to carry out the PWM controller specification on the basis of the average model of the PWM controlled system. This model is obtained by replacing the PWM regulator by means of a nonlinear memoryless saturation controller while leaving the system delays unaltered. Generally, the proposed design approach allows to totally circumvent the technical problems associated with the unbounded character of the PWM operator and the need for introducing low pass multipliers cascading the PWM controller ([2]).

In Section 2 the average PWM feedback controlled model is derived for general nonlinear dynamical systems described by delay differential equations. In Section 3 a satellite attitude control problem is presented. Section 4 contains the conclusions and suggestions for futher work.

2. DEFINITIONS AND BASIC RESULTS

2.1 Definition of Average PWM controlled model.

We shall distinguish three kinds of delays affecting the behavior of nonlinear systems. These are: plant delays , sensor delays and actuator delays.

Plant delays case

Let τ_p be a strictly positive constant. Consider a nonlinear delay differential system controlled by a PWM scheme based on the error signal. The system is assumed to be globally described in R^n, as :

$$
\begin{aligned}
& dx(t)/dt = f[x(t)] + \phi[x(t-\tau_p)] + g[x(t)]u \\
& y = h(x) \\
& e = y_d(t) - y \\
& u = M\ PWM_\tau[e(t_k)]
\end{aligned}
\tag{2.1}
$$

with f, g and ϕ globally smooth vector fields. The function h is a smooth scalar output function. The control input u is a discontinuous scalar control function obtained as the output of a Pulse-Width-Modulator excited by the error signal e. The sampling process is assumed to take place at regularly spaced time intervals of fixed duration T, i.e., $t_{k+1} = t_k + T$. M is a positive constant gain representing the maximum allowable input strength.

The PWM control operator, $PWM_\tau[e]$, is characterized by (See [2]):

$$
PWM_\tau[e(t_k)] = \begin{cases} \mathrm{sign}[e(t_k)] & \text{for } t_k \le t < t_k + \tau[e(t_k)]T \\ 0 & \text{elsewhere.} \end{cases}
\tag{2.2}
$$

where $\tau[e(t_k)]$ is known as the duty ratio function defined by :

$$
\tau[e(t_k)] = \begin{cases} \beta|e(t_k)| & \text{for } |e(t_k)| \le 1/\beta \\ 1 & \text{for } |e(t_k)| > 1/\beta \end{cases}
\tag{2.3}
$$

with β being a positive constant. Notice that :

$$
\tau[e(t_k)]\mathrm{sign}[e(t_k)] = \mathrm{sat}[e(t_k),\beta] := \begin{cases} \beta e(t_k) & \text{for } |e(t_k)| \le 1/\beta \\ \mathrm{sign}[e(t_k)] & \text{for } |e(t_k)| > 1/\beta \end{cases}
\tag{2.4}
$$

Proposition 2.1 As the sampling frequency F := 1/T tends to infinity, the description of the nonlinear controlled system (2.1) coincides with :

$$
\begin{aligned}
& dx(t)/dt = f[x(t)] + \phi[x(t-\tau_p)] + g[x(t)]v \\
& y = h(x) \\
& e = y_d(t) - y \\
& v = M\ \mathrm{sat}(e,\beta)
\end{aligned}
\tag{2.5}
$$

System (2.5) will be henceforth addressed as the average PWM controlled system.

Proof. Let $I_\phi(t)$ dentote the vector-valued function $\int_0^t \phi(x(\sigma))d\sigma$, with $\phi(x)$ being a given smooth vector field on R^n defining the flow of $dx/dt = \phi(x)$. Notice that the limit: $\lim_{T\to 0} [I_\phi(t+\alpha T)-I_\phi(t)]/T = \alpha\, d\, I_\phi(t)/dt = \alpha\, \phi(x(t))$. Similarly, consider $\phi[x(t-\tau)]$ for any fixed positive τ. We let $I_{\phi[x(t-\tau)]}(t) = \int_0^t \phi[x(\sigma-\tau)]d\sigma$. It follows that $\lim_{T\to 0} [I_{\phi[x(t-\tau)]}(t+T)-I_{\phi[x(t-\tau)]}(t)]/T = d\, I_{\phi[x(t-\tau)]}(t)/dt = \phi[x(t-\tau)]$. The sampled state-response of the nonlinear feedback system (2.1) can be written, in terms of an equivalent integral equation, as :

$$
\begin{aligned}
x(t_k+T) = x(t_k) + I_f(t_k+T) - I_f(t_k) + [I_{\phi[x(t-\tau_p)]}(t_k+T) - I_{\phi[x(t-\tau_p]}(t_k)] \\
+ M\, [\, I_g(t_k+\tau[e(t_k)]T) - I_g(t_k)]\mathrm{sign}\, [e(t_k)]
\end{aligned}
$$

Therefore, according to the above expression, the limit :

$$
\begin{aligned}
&\lim_{T\to 0,\, t_k\to t} T^{-1}[x(t_k+ T) -x(t_k)] =: dx(t)/dt \\
&= \lim_{T\to 0,\, t_k\to t} T^{-1}\{ [I_f(t_k+T)-I_f(t_k)] + [I_{\phi[x(t-\tau_p)]}(t_k+T)- I_{\phi[x(t-\tau_p)]}(t_k)] \\
&\qquad + M\, (I_g[t_k+ \tau[e(t_k)]T] - I_g(t_k)]\, \mathrm{sign}\, [e(t_k)]\ )\ \} \\
&= f[x(t)] + \phi[x(t-\tau_p)] + g[x(t)]\, M\, \tau[e(t)]\, \mathrm{sign}\, [e(t)] \\
&= f[x(t)] + \phi[x(t-\tau_p)]+ g[x(t)]\, M\, \mathrm{sat}[e(t),\beta] = f(x) + \phi[x(t-\tau_p)] + g(x)\, v \qquad \square
\end{aligned}
$$

Remark The behavior of the infinite frequency sampled delay differential system is then described by a nonlinear delay differential system with a continuous piece-wise linear control v, generated as the output of a memoryless nonlinear function of the saturation type. The saturation function, in turn, is excited by the error signal e. In other words, to evaluate the smooth average behavior of the actual PWM controlled system, the PWM controller is simply substituted by a nonlinear memoryless saturating controller independently of the plant delays.

The fundamental qualitative stability characteristics of the actual nonlinear PWM controlled system (2.1) are entirely captured by the average model (2.5). In order to show the validity of this assertion, it will be first analyzed the nature of the discrepancies among the open loop state trajectory responses of both system models, (2.1) and (2.5), when subject to the same arbitrary bounded measurable input signal. Later on, the close loop behavior will also be treated.

Let $e^*(t_k) = x(t_k) - z(t_k)$ be the discrepancy among the state x of the PWM controlled system (2.1) and the state z of the average system (2.5) at the sampling instant t_k. It will be assumed that the vector fields f(x) and ϕ are globally Lipschitz and that the vector field g(x) is globally bounded on R^n. i.e., there exist constants L_1 , L_2 and M such that $\|f[x(t)]-f[(z(t)]\| \le L_1 \|x(t)-z(t)\|$,

$\| \phi[x(t-\tau_p)]-\phi[z(t-\tau_p)] \| \le L_2\| x(t) - z(t) \|$ and $\|g(x)\| \le G$ for all x and z in R^n.

Theorem 2.2 Under the above assumptions on the vector fields f(x), $\phi(x)$ and g(x), given a small positive constant δ there exists, for any arbitrary finite time interval, $[0,NT]$, a sampling frequency $F_0 = 1/T_0$ such that, if the inital states discrepancy, $e^*(t_0) = x(t_0) - z(t_0)$, of systems (2.1) and (2.5) is norm bounded by δ, then the discrepancy $e^*(t_0+NT)$ of the corresponding state responses, due to the same arbitrary bounded measurable input r(t) with $|r(t)| \le K$ for $t \in [t_0,t_0 +NT]$, is norm bounded by $(1+\varepsilon)\delta$, for any sampling frequency $F > F_0$.

Proof According to (2.1) and (2.5) one has :

$$x(t_k+T) = x(t_k) + \int_{t_k}^{t_k+T} f(x(\sigma))d\sigma + \int_{t_k}^{t_k+T} \phi[x(\sigma-\tau_p)]d\sigma$$
$$+ M \int_{t_k}^{t_k+\tau[r(t_k)]T} g(x(\sigma)) \operatorname{sign} r(t_k)\, d\sigma$$
$$z(t_k+T) = z(t_k) + \int_{t_k}^{t_k+T} f(z(\sigma))d\sigma + \int_{t_k}^{t_k+T} \phi[z(\sigma-\tau_p)]d\sigma$$
$$+ M \int_{t_k}^{t_k+T} g(z(\sigma))\operatorname{sat}[r(\sigma),\beta]d\sigma$$

subtracting these expressions, one obtains :

$$e^*((t_k+T) = e^*(t_k) + \int_{t_k}^{t_k+T} \{f[x(\sigma)]-f[z(\sigma)]\}d\sigma + \int_{t_k}^{t_k+T}\{ \phi[x(\sigma-\tau_p)]- \phi[z(\sigma-t_p)]\}\, d\sigma$$
$$+ M \{ \int_{t_k}^{t_k+\tau[r(t_k)]T} g(x(\sigma)) \operatorname{sign} r(t_k)\, d\sigma - \int_{t_k}^{t_k+T} g(z(\sigma))\operatorname{sat}[r(\sigma),\, \beta]d\sigma \}$$

Hence:

$$\| e^*(t_k+T) \| \leq \| e^*(t_k)\| + L_1 \int_{t_k}^{t_k+T} \| e^*(\sigma) \|\, d\sigma + L_2 \int_{t_k}^{t_k+T} \| e^*(\sigma) \|\, d\sigma$$
$$+ M \| \int_{t_k}^{t_k+\tau[r(t_k)]T} g(x(\sigma)) \operatorname{sign} r(t_k)\, d\sigma - \int_{t_k}^{t_k+T} g(z(\sigma))\operatorname{sat}[r(\sigma),\beta]d\sigma \|$$
$$\leq \| e^*(t_k)\| + (L_1+L_2) \int_{t_k}^{t_k+T} \| e^*(\sigma) \|\, d\sigma$$
$$+ M \{ \int_{t_k}^{t_k+T} \| g(x(\sigma)) \|d\sigma + \int_{t_k}^{t_k+T} \beta \,\| g(z(\sigma)) \| |r(\sigma)|\, d\sigma \}$$
$$\leq \| e^*(t_k)\| + (L_1 + L_2) \int_{t_k}^{t_k+T} \| e^*(\sigma) \|\, d\sigma + M\,(1 + \beta K)\, T\, G$$

By the Gronwall-Bellman lemma one has, after letting $L_3 = L_1 + L_2$,

$$\|e^*(t_k+T)\| \leq [\, \| e^*(t_k) \| + M\,(1+\beta K)T\, G\,] \exp(\, L_3T\,)$$

Using iteratively the above inequality for $k=0,1,\ldots,N$, the following loose estimate for the norm of the discrepancy $e^*(t_0 + NT)$ is easily obtained,

$$\| e^*(t_0 + NT) \| \leq [\, M(1+\beta K)NG\, T + \| e^*(t_0) \|\,] \exp(\, L_3NT\,)$$

i.e., the discrepancy, at an arbitrary finite time $t_0 + NT$, among the actual open loop PWM state response and that of the average system remains bounded by an amount determined by the initial states discrepancy, the system constants and the sampling interval T. The following trascendental equation, is easily obtained from the previous equation:

$$(1+\varepsilon)\, \delta \exp\{\, -L_3N\, T\, \} = [\, M(1+\beta K)NG\, T + \delta]$$

It is easy to see that because the right hand side term of this equation monotonically increases with T from the value δ at $T = 0$, while the left hand side term monotonically decreases with T from the value $(1+\varepsilon)\delta > \delta$ at $T = 0$, it has a unique solution for some $T = T_0 > 0$. Hence, given an initial discrepancy bound, $\| e^*(t_0) \| \leq \delta$, and a small positive constant ε, a sampling frequency $F_0 = 1/T_0$ exists for which a preassigned error response bound Δ can be obtained at any finite later time $t_0 + NT$, $\| e^*(t_0+NT) \| \leq \Delta$, with Δ being of the form $(1+\varepsilon)\delta$. It follows that for any sampling frequency $F > F_0$ (i.e., $T < T_0$) a states discrepancy strictly bounded by $(1+\varepsilon)\delta$ is obtained among the open loop state responses at time $t_0 + NT$. □

The qualitative stability characteristics of the delay differential PWM controlled system (2.1) are captured by its smooth infinite-frequency sampling average model (2.5). In particular, a necessary condition for asymptotic stability of the actual PWM system (2.1) to the zero error manifold, is the asymptotic stability of the average model to such surface. It is also obvious that if the response of the average model (2.5) is asymptotically stable to $e = 0$, the actual PWM controlled response can be made to follow, arbitrarily close, the response of the average model by suitably increasing of the sampling frequency in the PWM controller. Thus, modulo sufficiently large -but finite- sampling frequency, the following theorem holds true.

Theorem 2.3 The closed loop PWM controlled system (2.1) is asymptotically stable to the $e = 0$ manifold if and only if the average PWM system (2.5) is asymptotically stable to such manifold.

Proof The average model of the PWM controller establishes three disctintive regions in the state space of the system. These are : the saturation regions $S_{+1} = \{ x : e = -h(x) > 1/\beta$ and $S_{-1} = \{ x : e = - h(x) < -1/\beta \}$ and the linearity, or boundary layer region $S\beta = \{ x : -1/\beta \leq e = -h(x) \leq 1/\beta \}$.

In the saturation regions the actual delay differential PWM controlled system (2.1) totally coincides with the average model (2.5). Hence, under exactly the same functional and static initial conditions defined, respectively in $-\tau_p \leq t < 0$ and $t = 0$, the actual PWM system trajectories and those of the average PWM controlled system entirely coincide. By virtue of a straightforward extension of the continuity of solutions in the initial conditions, it follows that for arbitrarily small discrepancies of the initial state functions or initial states, the corresponding state trajectories of the actual and the average PWM systems remain arbitrarily close to each other in such regions.

Within the boundary layer region, the actual PWM system is described by a two position switch (ON-OFF type). Depending on the sign of the error signal the PWM controlled system is described as follows:

For $e > 0$:

$$dx/dt = f(x) + \phi[x(t-\tau_p)] + u\, g(x) ; \qquad u = \begin{cases} +1 & \text{for } t_k < t \leq t_k + \tau(e)T \\ 0 & \text{elsewhere} \end{cases}$$

For $e < 0$:

$$dx/dt = f(x) + \phi[x(t-\tau_p)] + u\, g(x) \quad ; \quad u = \begin{cases} -1 & \text{for } t_k < t \leq t_k + \tau(e)T \\ 0 & \text{elsewhere} \end{cases}$$

It can be shown that these systems exhibit global sliding motions about integral manifolds of the average PWM controlled system (which in fact corresponds to the ideal sliding dynamics of the equivalent sliding motion). It follows that, within the boundary layer region, the trajectories of the average PWM controlled system are followed, arbitrarily close, by the actual PWM controlled responses in the same manner that a sliding motion follows, on a given surface in the state space, the trajectories of the corresponding ideal sliding dynamics. Increasing of the sampling frequency in the PWM scheme is totally equivalent to discarding imperfections of the equivalent variable structure switch. In the limit, PWM controlled trajectories would coincide with its average description in the same manner that an actual sliding regime coincides with its ideal sliding dynamics. □

Actuator and Sensor Delays

Let τ_a and τ_s be positive constants. We say that (2.1) has actuator delays whenever the control function u is of the form $u(t_k+\tau_a) = M\ \mathrm{PWM}[e(t_k)]$. The controlled system is said to have sensor delays if $e(t) = y_d(t) - y(t-\tau_s)$. It is easy to show that actuator delays and sensor delays, similarly to the case of plant delays, do not affect the form of the average PWM controller. In such cases, the PWM controller is still substituted by the saturation nonlinearity while the blocks representing, or containing, time delays remain unaltered in the average model. This is easily seen to be a valid statement from the following straightforward considerations.

Sensor delays : Let

$$e(t_k) = y_d(t_k) - y(t_k-\tau_s) \quad \text{i.e.,} \quad e(t) = y_d(t) - y(t-\tau_s)$$

then

$$\lim_{T\to 0,t_k\to t} \{ T^{-1} M\, [I_g(t_k+\tau[e(t_k)]T) - I_g(t_k)] \}\ \mathrm{sign}[e(t_k)]$$

$$= M\,\tau[\ y_d(t) - y(t-\tau_s)\ ]\ g(x(t))\ \mathrm{sign}\ [\ y_d(t) - y(t-\tau_s)\ ]$$

$$= M\,\tau[e(t)]\ g(x(t))\ \mathrm{sign}\ [\ e(t)\ ]\ = M\ \mathrm{sat}(e(t),\beta\ )\ g(x(t)) = g(x(t))v(t)$$

i.e., the form of the average controller is unaffected by the fact that sensor delays are present in the feedback loop.

Actuator delays : In this case, the control function is given by

$$u(t) = \mathrm{PWM}\ _\tau[e(t_k-\tau_a)] = \begin{cases} \mathrm{sign}[e(t_k-\tau_a)] & \text{for}\ \ t_k \le t < t_k + \tau[e(t_k-\tau_a)]T \\ 0 & \text{elsewhere.} \end{cases}$$

hence,

$$\lim_{T\to 0,t_k\to t} T^{-1}\int_{t_k}^{t_k+T} M\ g(x(\sigma))u(\sigma)d\sigma$$

$$=\lim_{T\to 0,t_k\to t} T^{-1}\ M \int_{t_k}^{t_k+\tau[e(t_k-\tau_a)]T} g(x(\sigma))d\sigma\ \mathrm{sign}[e(t_k-\tau_a)]$$

$$= \lim_{T\to 0,t_k\to t} T^{-1}\ M \int_{t_k-\tau}^{t_k-\tau_a+\tau[e(t_k-\tau_a)]T} g(x(\sigma+\tau_a))d\sigma\ \mathrm{sign}[e(t_k-\tau_a)]$$

$$= M\,\tau[e(t-\tau_a)]g(x(t))\mathrm{sign}[e(t-\tau_a)] = g(x(t))\ M\ \mathrm{sat}[e(t-\tau_a),\beta] = g(x(t))v(t-\tau_a)$$

From the second line to the third line above, a change of the time integration variable σ by $\sigma-\tau_a$ was performed while noting that the duty ratio, $\tau[e(t_k-\tau_a)]$, is a fixed constant during the integration period and precisely determined at time $t_k-\tau_a$. The average model of the PWM controller in a system subject to actuator delays is still a memoryless saturation type of controller excited by delayed errors. It is easy to see that the delay operator and the saturation nonlinearity commute with each other. The result follows. □

2.3 Linear Controlled Plants

For the case of linear controlled plants without delays, a vast amount of input-output design methods can be directly used for PWM controller design based on the average PWM model. The design problem is reduced to finding the stabilizing gain β corresponding to the linear part of the saturation block in the classical feedback configuration shown in Figure 1. Among the available methods to solve such a problem one finds : the Small Gain Theorem, the Circle Criterion, Describing Function methods, Nyquist stability criterion. For linear plants including finite time delays, the Popov criterion is especially suitable for carrying out an input-output stabilizing desing based on the average PWM controlled model. All these classical design techniques, readily found in the literature, are based on well-known sufficient conditions for stability and asymptotic stability of linear feedback controlled plants.

Example. Consider the problem of designing a stabilizing PWM controller for a distributed linear plant described by :

$$G(s) = e^{-Ts}(s+p)^{-1} \quad ; \quad p > 0$$

Let g_{av}(PWM) denote the gain of the average PWM controller. Then, $g_{av}(PWM) = \sup_{e \neq 0} |sat(e,\beta)||e|^{-1} = \beta$. The Small Gain theorem leads to the following sufficient condition for asymptotic stability :

$$g_{av}(PWM).\sup_{w} | e^{-Tjw}(p+jw)^{-1} | = \beta p^{-1} < 1 \Rightarrow \beta < p$$

One may instead use a particularization of the Circle Criterion for the design case of the saturation controller. This criterion leads to :

$$\inf_{w} \mathrm{Re} \, [\, e^{-jwT}(p+jw)^{-1} \,] > -\beta^{-1}$$

3.APPLICATIONS TO A SATELLITE ATTITUDE CONTROL PROBLEM

Here we consider a single-axis attitude control problem for a well known linearized benchmark model representing a satellite plant (the model is taken from Howe and Cavanaugh [6]). This model includes fast dynamical rate and position sensors as well as a first order model for the actuator preceeded by a small pure time delay representing the gas jet reaction control system. The pitch angle θ and the pitch rate $d\theta/dt$ are used in a PWM feedback scheme designed to follow a desirable command angle θ_r. The PWM controlled model is shown in a block diagram form in Figure 2. We assume that the reference pitch angle θ_r is 0.09 radians.

The PWM controller design entitles the specification of the constant β and the sampling frequency 1/T. A satisfactory design, based on the average PWM controlled model, necessarily implies the use of high frequency sampling if basic qualitative and quantitative features of the average designed response are to be captured by the actual PWM controlled process, within arbitrarily small discrepancies. The estimation of the necessary sampling frequency can be assessed from the results of theorem 2.2 above. However, this issue, more related to response precision, can be easily handled by extensive simulation and, thus, it will not concern us here. The more relevant task is, in our view, constituted by the choosing of the constant β, due to its direct influence on the stability of the average closed loop system.

For the case at hand, the closed loop system is guaranteed to be asymptotically stable, according to a well known particularization of Sandberg's Circle Criterion, if

$$-1/\beta < \inf_w \mathrm{Re}[\, G(jw)]$$

where $G(jw)$ represents the complex transfer function of the single input-single output open loop system, including the actuator dynamics, actuator delay, and sensor dynamics.

Figure 3. shows a Nyquist diagram of the open loop system for a normalized PWM slope gain of $\beta = 1$. Any positive value of β smaller than, say, 1500 guarantees asymptotic stability. Simulated asymptotically stable plant state average responses are shown in Figure 4 corresponding to $\beta = 50$. The actual PWM responses do not differ significantly from the average responses due to the low pass filter effect of the actuator dynamics.

4. CONCLUSIONS AND SUGGESTIONS FOR FURTHER RESEARCH

In this article a design procedure is proposed for the specification of stabilizing feedback PWM controllers in linear and nonlinear delay differential systems. It is shown that, under an infinite sampling frequency assumption, an average model for the classical PWM ON-OFF-ON controller is obtained by simply replacing the discontinuous PWM controller by a memoryless saturation type of compensator. This average model was shown to capture the basic qualitative (i.e. stability) features of the actual PWM controlled system. This allows to treat the PWM design problem in a cleaner fashion than using traditional discrete-time approximations. At the same time one totally circumvents the technical difficulties associated to the fact that PWM operators are indeed unbounded operators on the Banach space of absolutely integrable functions. For the particular case of linear controlled plants, the results of this paper make readily suitable for PWM controller design, a vast number of traditional input-output design techniques, based on well-known sufficient conditions for asymptotic satability. The results were applied to PWM controller specification for a seventh order linearized satellite plant, including actuator delays, in a pitch angle attitude control problem of the reorientation type.

The results of the article can be easily extended to multivariable PWM controllers with, or without, delays. Connections arise among high-gain systems and nonlinear average PWM controlled systems when the saturation nonlinearity, replacing the PWM controller, exhibits a large slope. This also explains some equivalences among high gain controlled systems and variable structure systems undergoing sliding motions. Interesting connections may be expected, for the case of linear PWM controlled multivariable plants, with H_∞ control theory.

REFERENCES

[1] Ya.Z. Tsypkin. *Relay Control Systems*, Cambridge University Press, Cambridge, 1984.

[2] R.A. Skoog and G.L. Blankenship, " Generalized Pulse-Modulated Feedback Systems : Norms, Gains, Lipschitz Constants, and Stability," *IEEE Trans. on Auto. Control*, Vol. AC-15, No.3, pp. 300-315, June 1970.

[3] M. La Cava, G. Paletta and C. Picardi, "Stability Analysis of PWM Feedback Control Systems with PID Regulators," *Int. J. of Control*, Vol. 39, No. 5, pp. 987-1005. May 1984.

[4] H. Sira-Ramirez, "Sliding Mode Control on Slow Manifolds of DC-to-DC Power Converters," *Int. Journal of Control*, Vol. 47, No. 5 pp 1323-1340. 1988

[5] H. Sira-Ramirez, " A Geometric Approach to Pulse-Width-Modulated Control in Nonlinear Dynamical Systems," *IEEE Trans. on Auto. Control.* January 1989.

[6] R.M Howe and W.P. Cavanaugh, " AD 100 Simulation of a Satellite Attitude Control System," Application Report. Applied Dynamics International, 3800 Stone School Road. Ann Arbor, MI 48104. 1985.

FIGURES

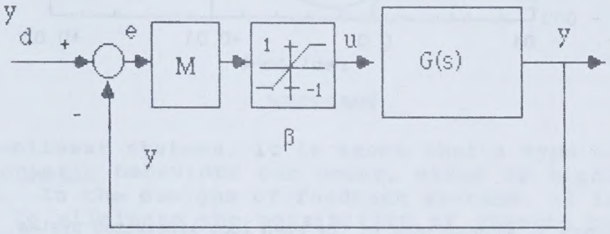

y_d
+
e
M
1
-1
u
G(s)
y
-
β
y

Figure 1. Average Model of Linear Plant Controlled by Pulse-Width-Modulation

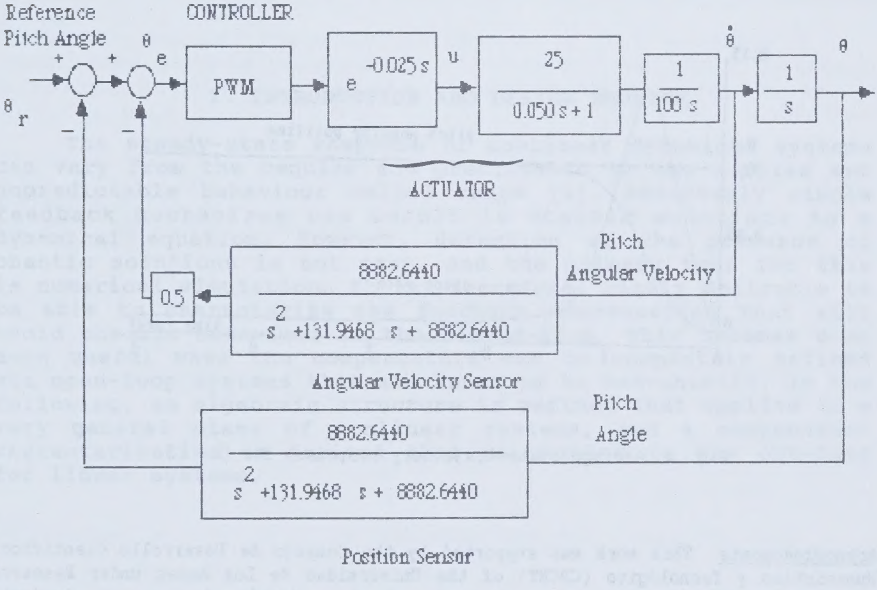

Reference
Pitch Angle
CONTROLLER
+
+
θ
e
PWM
$e^{-0.025\,s}$
u
$\frac{25}{0.050\,s+1}$
$\dot{\theta}$
$\frac{1}{100\,s}$
$\frac{1}{s}$
θ
θ_r
−
−
ACTUATOR
Pitch
Angular Velocity
0.5
$\frac{8882.6440}{s^2+131.9468\,s+8882.6440}$
Angular Velocity Sensor
Pitch
Angle
$\frac{8882.6440}{s^2+131.9468\,s+8882.6440}$
Position Sensor

Figure 2. A PWM Satellite Atttitude Feedback Control System

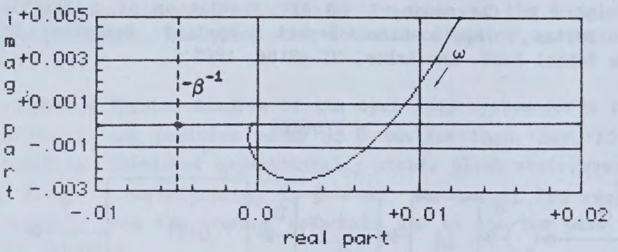


Figure 3. Nyquist plot of the open loop linearized system (PWM operator gain, β, normalized to the value of 1)

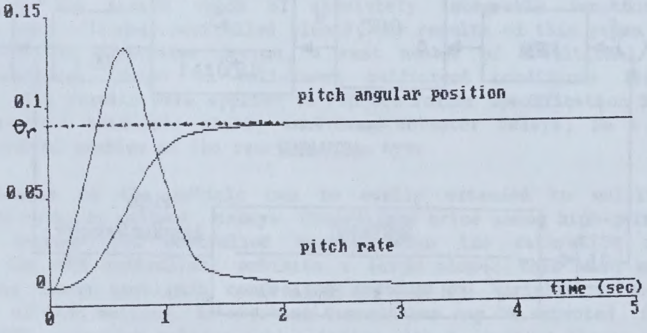


Fig. 4. Average state trajectory responses for β = 50

Acknowledgments This work was supported by the Consejo de Desarrollo Científico, Humanístico y Tecnológico (CDCHT) of the Universidad de Los Andes under Research Grant I-280-87. The generous support of Dr. Julián Aguirre (Academic Vice-President of the University) and Dr. Juan Puig (General Chairman of the CDCHT) in making possible the presentation of this article is gratefully acknowledged.

DESIGN OF NONCHAOTIC FEEDBACK SYSTEMS

VIJAY RAMAN
Department of Electrical Engineering
Florida International University
Miami, Florida, 33199

and

RUEY-WEN LIU
Department of Electrical and Computer Engineering
University of Notre Dame
Notre Dame, IN 46556

ABSTRACT

In nonlinear systems, it is known that a type of response known as chaotic behaviour can occur, which is highly complex in nature. In the designs of feedback systems, it is normally essential to eliminate the possibility of chaotic behavior in any part of the system, as a first condition in the design objective. This problem is considered in this paper. Necessary and sufficient conditions are shown for the choice of compensators to satisfy design objectives while eliminating chaos, for a very general class of systems.

I. INTRODUCTION AND DESIGN PROBLEM

The steady-state response of nonlinear dynamical systems can vary from the regular and predictable to the complex and unpredictable behaviour called chaos [1]. Reasonably simple feedback mechanisms can result in chaotic solutions to a dynamical equation. However, detection of the presence of chaotic solutions is not easy, and the primary tool for this is numerical simulation. It is, therefore, highly desirable to be able to characterize the feedback compensation that will avoid chaotic behaviour in the closed-loop. This becomes even more useful when the compensators can be completely defined via open-loop systems that are known to be non-chaotic. In the following, an algebraic structure is defined that applies to a very general class of nonlinear systems, and a compensator characterization is derived that is similar to one obtained for linear systems.

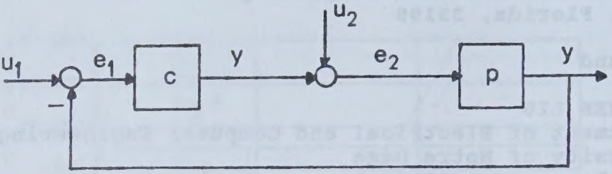


Figure 1

In Figure 1, P and C represent the nonlinear plant and nonlinear compensator respectively. The signals u_1, u_2, e_1, e_2, y_1, y_2, are all assumed to lie in a normed linear vector space V. Thus P and C are maps from V to V. This assumption implies that for multi-input-multi-output (MIMO) systems, the number of inputs and outputs must be equal for both P and C; however, this assumption is for convenience, and the extension to the general case can be carried out by the use of projection operators [4].

A subspace U of V is used to represent the set of "good" signals. From the viewpoint of eliminating chaotic behaviour, U may reasonably be given by the set of almost-periodic signals.

A system G: V --> V is called nonchaotic if there exist constants denoted by K_G and L_G such that for any u in U with $||u|| < K_G$, Gu is in U and $||Gu|| < L_G||u||$. Thus bounded "good" inputs are mapped by nonchaotic systems to "good" outputs, with finite gain.

The goals of the feedback design for the system of Figure 1 are stated in the following:

Feedback Design Problem (FDP) - Given P, to obtain C such that the following two conditions are satisfied:

(i) there exist constants K_1 and K_2 such that for all u_1, u_2 in U with $||u_1|| < K_1$, and $||u_2|| < K_2$, the signals e_1, e_2, y_1 are in y_2 are in U, and

(ii) if $u_2=0$, the the operator H_{11}: u_1 --> y_1 satisfies some specified design condition 'D'.

Condition (i) of the FDP requires the elimination of chaotic behavior throughout the system, while condition (ii) deals with specified behaviour from the primary input to primary output. The following theorem gives the main result regarding the complete solution to the FDP.

II. SOLUTION TO DESIGN PROBLEM

Theorem: Let the plant P be non-chaotic and possess the additional property that there exist two associated constants A_P and B_P such that for all v in V and all u in U with $||u|| < A_P$, it follows that $P(v+u) - Pv$ is in U and $||P(v+u) - Pv|| < B_P ||u||$. Then, a solution C to the feedback design problem exists if and only if there exists nonchaotic Q with the operator (I-PQ) being invertible and the operator PQ satisfying design condition D. Furthermore, the set of all C'S solving the feedback design problem, is generated by the set of all nonchaotic Q's satisfying the above requirements, via the relationship $C = Q(1-PQ)^{-1}$.

Proof:

(Sufficiency, or Q => C):

By hypothesis for this part, consider a nonchaotic Q such that $(I-PQ)^{-1}$ exists and PQ satisfies design condition D. Then define

$$C = Q(I-PQ)^{-1} \tag{1}$$

It is easily shown that this implies $Q = C(I+PC)^{-1}$.

The proof will demonstrate by a specific choice the existence of K_1 and K_2 such that if u_1, u_2 are in U with $||u_1|| < K_1$ and $||u_2|| < K_2$, then e_1, e_2, y_1, y_2 are in U. Accordingly, let

$$\begin{aligned} K_1 &= 1/2 \ \mathrm{Min}(\ K_Q,\ K_P/L_Q\ ) \\ K_2 &= \mathrm{Min}(\ A_P,\ \mathrm{Min}(K_Q,\ K_P/L_Q)/2B_P\ ) \end{aligned} \tag{2}$$

Define the signal j,

$$\begin{aligned} j &= (I+PC)e_1 \\ &= e_1 + PCe_1 \end{aligned} \tag{3}$$

Now,

$$\begin{aligned} e_1 &= u_1 - y_1 = u_1 - P(u_2+y_2) \\ &= u_1 - P(u_2+Ce_1) \end{aligned}$$

Therefore,

$$j = u_1 - P(u_2+Ce_1) + P(Ce_1) \tag{4}$$

Since $||u_2|| < K_2 < A_P$ by the choice of K_2, the property of P specified in the statement of the theorem implies that

$$P(u_2+Ce_1) - P(Ce_1) \text{ is in U, and}$$

$$||P(u_2+Ce_1) - P(Ce_1)|| < B_P\,||u_2||$$

Therefore, since $||u_1|| < K_1$ and $||u_2|| < K_2$,

$$j \text{ is in U, and } ||j|| < K_1+B_PK_2 \tag{5}$$

Using (3),

$$e_1 = (I+PC)^{-1}j = (I-PQ)j$$

$$= j- PQj \tag{6}$$

Now,

$$\begin{aligned} ||j|| &< K_1 + B_PK_2 \\ &< K_Q/2 + B_P.K_Q/2B_P \\ &= K_Q \end{aligned}$$

by using (2). Since Q is nonchaotic,

$$\begin{aligned} ||Qj|| &< L_Q\,||j|| \\ &< L_Q(K_1+B_PK_2) \\ &< L_Q(K_P/2L_Q + B_P.\,(K_P/L_Q)/2B_P) \\ &< K_P/2 + K_P/2 \\ &= K_P \end{aligned}$$

by using (2). Therefore, since P is nonchaotic,

$$PQj \text{ is in U, and}$$

$$||PQj|| < L_P\,||Qj|| < L_PK_P$$

Since j and PQj are in U and bounded, (6) shows that

$$e_1 \text{ is in U, and is bounded.} \tag{7}$$

Now,

$$y_2 = Ce_1$$
$$= C(I-PQ)j$$
$$= Q(I-PQ)^{-1}(I-PQ)j \quad \text{by (1)}$$
$$= Qj \qquad (8)$$

which has been established to be in U, and bounded.

It immediately follows that

$$y_1 = u_1 - e_1 \quad \text{is in U and bounded} \qquad (9)$$

and

$$e_2 = u_2 + y_2 \quad \text{is in U and bounded.} \qquad (10)$$

(7),(8),(9) and (10) show that the defined compensator C satisfies condition (i) of the feedback design problem.

Furthermore, if $u_2=0$,

$$y_1 = Py_2$$
$$= PQj \quad \text{by (8)}$$

But (4) implies that $j = u_1$ when $u_2=0$. Therefore,

$$y_1 = PQu_1,$$

and the operator from u_1 to y_1 is given by $H_{11} = PQ$, and therefore satisfies the design condition 'D' by the hypothesis of the theorem. This shows that C also satisfies condition (ii) of the feedback design problem. This completes the proof for this part (sufficiency) of the theorem.

(Necessity, or C => Q):

According to the hypothesis for this part, let the compensator C and constants K_1 and K_2 exist such that for all u_1 and u_2 in U with $||u_1|| < K_1$ and $||u_2|| < K_2$, the signals e_1, e_2, y_1 and y_2 are in U and bounded, and furthermore, if $u_2=0$, then the operator H_{11}: $u_1 \dashrightarrow y_1$ satisfies design condition 'D'.

Define

$$Q = C(I+PC)^{-1} \qquad (11)$$

so that $C=Q(I-PQ)^{-1}$. The relationship between C and Q is as before.

It is required to first show that Q is nonchaotic. In order to show this, let $u_2=0$. Then,

$$y_2 = Qj \qquad \text{as shown in (8)}$$

Also $j = u_1$ when $u_2=0$, from (4). Thus,

$$y_2 = Qu_1 \tag{12}$$

For u_1 in U and $||u_1|| < K_1$, y_2 is in U and bounded, by hypothesis. Hence, define

$$K_Q = K_1 \tag{13}$$

$$L_Q = \underset{\substack{||u_1|| < K_1 \\ ||u_1|| = 0}}{\text{supremum}} \left(||y_2||/||u_1||\right) \tag{14}$$

Then, for all u_1 in U with $||u_1|| < K_Q$,

$$Qu_1 = y_2 \quad \text{is in U, and}$$

$$||Qu_1|| = ||y_2|| < L_Q \, ||u_1|| \tag{15}$$

which shows that Q is indeed a nonchaotic operator.

Furthermore, as in the sufficiency proof, when $u_2=0$,

$$y_1 = Py_2$$

$$= PQu_1$$

so that PQ is the operator H_{11}: $u_1 \dashrightarrow y_1$. The fact that C is a solution to the feedback design problem shows that PQ satisfies design condition 'D'. This complete the proof of the theorem.

This theorem is related to earlier stabilization results [2-4] and utilizes the properties of nonchaotic behavior for bounded signals. The algebraic view point therein is to show that parallel and cascade combinations of nonchaotic systems are also nonchaotic, in fact nonchaotic systems form a left distributive ring. It is then established that all transfer operators in the feedback system will be nonchaotic if and only if one specific transfer operator Q is nonchaotic.

Remark 1: It should be noted that a nonchaotic system as defined here will not exhibit chaotic behavior only provided the inputs are appropriately bounded.

Remark 2: The invertibility condition of Theorem 1 specifies the existence of the inverse of (I-PQ) as a nonlinear operator, but it need not be nonchaotic.

Remark 3: Nonlinear systems that can be described by Volterra kernels satisfy a "nonchaotic" condition as given here [5], that is, almost-periodic inputs, if appropriately bounded, give rise to only almost-periodic outputs. Therefore, if the plant P has a Volterra kernel and satisfies the condition of the Theorem on its incremental behavior, choosing any Q that has a Volterra kernel (and satisfies the invertibility condition) will guarantee a.p. responses to appropriately bounded a.p. inputs.

III. CONCLUSIONS

Elimination of chaotic behaviour due to feedback in nonlinear systems has been considered here. A normed signal space setting has been used in order to embed the key condition that "good" or nonchaotic behaviour can be guaranteed up to a certain bound on the magnitudes of the inputs. An equivalence between the feedback compensator and an open-loop "nonchaotic" operator has then been shown, which defines the class of compensators that eliminate closed-loop chaotic behaviour. The result is quite general, and not limited to specific nonlinearities.

An interesting condition that arises is that the open-loop plant must exhibit incrementally nonchaotic behaviour, that is, a small nonchaotic change in the input should give only a small nonchaotic change in the output. The meaning of this for specific nonlinearities requires investigation.

REFERENCES

[1] T.A. Parker and L.O. Chua, "Chaos: A Tutorial for Engineers," Proceeding of the IEEE, Vol. 75, No. 8, pp. 982-1008, August 1987.

[2] C.A. Desoer and R.W. Liu, "Global Parametrization of Feedback Systems," Memorandum No. UCB/ERL/M81/57, Electronics Research Laboratory, College of Engineering, University of California, Berkeley, July 24, 1987.

[3] R. Liu and C.H. Sung, "Linear Feedback System Design", Circuits, Systems and Signal Processing, Vol. 2, No. 1, pp. 35-44, 1983.

[4] R. Liu and C.H. Sung, "Design of Nonlinear Feedback Systems: An Algebraic Theory", Proceedings IACAS 1983.

[5] S. Boyd, L.O. Chua and C.A. Desoer, "Analytical Foundations of Volterra Series", IMA Journal of Mathematical Control and Information (1984) 1, pp 243-282, 1984.

Liapunov Functions
for
Critical Nonlinear Systems

Jyun-Horng Fu and Eyad H. Abed
Department of Electrical Engineering
University of Maryland, College Park, MD 20742 USA

September 27, 1988

Abstract

Liapunov functions for critical nonlinear systems are constructed and relate to stability criteria for bifurcations. Specifically, we show that stability can be determined from the quadratic and cubic parts of the dynamics restricted to the tangent space associated with the critical mode. We also present a new derivation of stability criteria of bifurcated solutions generalizing a result of Mees and Chua for planar systems at a Hopf bifurcation point.

1 Introduction

It is often cited that stability of bifurcated solutions, in particular bifurcation formulae, can be derived via constructing an appropriate Liapunov function for the equilibrium at criticality. This is in fact justified by virtue of the center manifold theorem. However, Liapunov functions thus constructed apply only to either the scalar or planar systems resulting from center manifold reduction. For Hopf bifurcation, the resulting formula consists of ten terms, which is clearly not convenient especially when stabilizing feedback control is sought. Simpler formulae have been obtained through a projection method employing Taylor series expansion [4,5,8]. In this paper, the corresponding Liapunov functions are constructed directly for the given dynamics. This approach is promising for the estimation of domains of attraction.

Consider the following nonlinear dynamical system

$$\dot{x} = f(x) = L_0 x + Q_0(x, x) + C_0(x, x, x) + \cdots \tag{1}$$

where $f(0) = 0$ and $x \in \mathbb{R}^n$. Here, $L_0 x$ is the linear part while $Q_0(\cdot,\cdot)$ and $C_0(\cdot,\cdot,\cdot)$ are the quadratic and cubic parts in the Taylor expansion of $f(\cdot)$ near $x = 0$. These arise from symmetric bilinear and trilinear forms, respectively. For example, $Q_0(x,y) = Q_0(y,x)$. We are interested in two cases: $0 \in \sigma(L_0)$ (Case S) or, $\pm i\omega_c \in \sigma(L_0)$ (Case H), and in either case the rest of the spectrum lies entirely on the open left half complex plane. Denote eigenvalues of L_0 by λ_i with left and right eigenvectors l_i and r_i, where $\Re e(\lambda_i)$ decreases with i. If any of eigenvalues of L_0 lie on the imaginary axis, then linear theory fails in ascertaining local stability properties of Eq. (1). These cases are known as critical cases in stability, and clearly Case S and Case H are the two simplest possible cases.

Remark 1. Case S corresponds to *stationary bifurcation* while Case H corresponds to *Hopf bifurcation* of an associated one-parameter family of differential equations [1-6].

2 Liapunov Functions for Critical Systems

Mees and Chua [6] use a Liapunov function containing quadratic, cubic, and quartic terms to arrive at a formula for β_2 for Hopf bifurcation. It turns out that for the n-dimensional systems no quartic terms are really necessary (this will be explained below). Hence we shall seek a Liapunov function $V(x)$ for the critical system (1), consisting of the sum of a quadratic and cubic part, namely,

$$V(x) = x^T P x + K(x,x,x). \tag{2}$$

For $V(x)$ to be locally positive definite it is natural to require P be symmetric and positive definite. In addition, $K : (\mathbb{R}^n)^3 \to \mathbb{R}$ is a symmetric cubic form which obviously cannot be sign-definite.

The time derivative $\dot{V}(x)$ of $V(x)$ along the trajectories of (1) can be written as

$$\begin{aligned} \dot{V}(x) &= x^T(L_0^T P + P L_0)x \\ &\quad + x^T P Q_0(x,x) + Q_0^T(x,x) P x + 3K(L_0 x, x, x) \\ &\quad + x^T P C_0(x,x,x) + C_0^T(x,x,x) P x + 3K(Q_0(x,x),x,x) + \cdots \end{aligned} \tag{3}$$

If we can construct suitable P and $K(\cdot,\cdot,\cdot)$ such that $-\dot{V}(x)$ is locally positive definite, then we may conclude that the origin of (1) is locally asymptotically stable, as a standard result of Liapunov stability theory. The reason for including a cubic form $K(x,x,x)$ in the Liapunov function is simple. Indeed, if L_0 were Hurwitz stable, then for any $Q > 0$ the associated Liapunov equation $L_0^T P + P L_0 = -Q$ has a unique positive definite solution P. In critical cases, it may be that no pair of positive definite matrices P, Q satisfying the Liapunov matrix equation exists. Hence the stability of a nonlinear critically stable system cannot always be decided by a Liapunov function consisting of a quadratic form.

We proceed now to the derivation of Liapunov function for Case S. The derivation for Case H is similar and so is presented only briefly.

2.1 One Zero Eigenvalue

Decompose x into two components by writing x as

$$x = ar + w := ar + \sum_{k=2}^{n} a_k r_k \in \mathbb{R}^n \tag{4}$$

where $r := r_1$, $a \in \mathbb{R}$, $ar \in E^c := \text{span}\{r\}$ and $w \in E^s := \mathbb{R}^n \setminus E^c$, with $lw = 0$, lies in the subspace spanned by the eigenvectors associated with the stable modes of L_0. If $\Im m(\lambda_k) \neq 0$, $\bar{a}_k = a_k \in \mathbb{C}$. Moreover, we have the bi-orthogonality relations shared among the eigenvectors, namely,

$$l_i r_j = \delta_{ij} \tag{5}$$

with normalization $l_i r_i = 1$ [5]. By virtue of bi-orthogonality we hypothesize the form

$$P = l^T l + \sum_{k=2}^{n} c_k l_k^T l_k := l^T l + Z, \; c_k > 0 \tag{6}$$

where $l := l_1$, for P. This is an example of a dyadic form which differs from the more common dyadic form $\sum c_k r_k l_k$ [9]. For nonreal λ_k the corresponding two terms are modified into $\bar{l}_k^T l_k + l_k^T \bar{l}_k$ sharing the same coefficient c_k so as to accommodate the fact that P is real and symmetric. Note that we have normalized $c_1 = 1$ and that c_k must be positive.

Observe that

$$x^T(L^T P + PL)x = \sum_{k=2} c_k a_k^2 \Re e(\lambda_k) \tag{7}$$

which clearly vanishes for all $x \in E^c$ (since $w = 0$) and is negative otherwise. This is related to the fact that linearization of (1) fails to determine stability.

Substituting $x = ar + w$ in $\dot{V}(x)$ in Eq. (3) and collecting like powers of $|a|$ and $||w||$, we have

$$\dot{V}(ar + w) := \dot{V}_2 + \dot{V}_3 + \dot{V}_4 + \cdots \tag{8}$$

with the subscript denotes the order of a term with respect to $|a|$ and $||w||$. For instance, $\dot{V}_2$, $\dot{V}_3$, and $\dot{V}_4$ are given by

$$\dot{V}_2 \;=\; w^T(L_0^T Z + Z L_0)w < 0, \tag{9}$$

$$
\begin{aligned}
\dot{V}_3 \;=\;& 2a^3 lQ_0(r,r) + 4a^2 lQ_0(r,w) + 2alQ_0(w,w) \\
&+ 2a^2 w^T ZQ_0(r,r) + 4aw^T ZQ_0(r,w) \\
&+ 3a^2 K(L_0 w, r, r) + 6aK(L_0 w, w, r) \\
&+ 2w^T ZQ_0(w,w) + 3K(L_0 w, w, w),
\end{aligned} \tag{10}
$$

and,

$$
\begin{aligned}
\dot{V}_4 \;=\;& a^4[2lC_0(r,r,r) + 3K(Q_0(r,r),r,r)] \\
&+ 2a^3[3lC_0(r,r,w) + 3K(Q_0(r,r),r,w) + 3K(Q_0(r,w),r,r)] \\
&+ 3a^2[2lC_0(r,w,w) + K(Q_0(w,w),r,r) \\
&\qquad + 4K(Q_0(r,w),r,w) + K(Q_0(r,r),w,w)] \\
&+ 2a[lC_0(w,w,w) + 3K(Q_0(w,w),r,w) + 3K(Q_0(r,w),w,w)] \\
&+ [2w^T ZC_0(w,w,w) + 3K(Q_0(w,w),w,w)].
\end{aligned} \tag{11}
$$

Note that Eq. (9) implies that $\dot{V}_2 < 0$ whenever $w \neq 0$, $w \in E^s$. Also note that of the terms appearing in Eq. (10), only $2w^T ZQ_0(w,w)$ and $3K(L_0 w, w, w)$ are dominated by $\dot{V}_2$.

At this point it may appear that we can impose conditions on the coefficients of the terms with various powers of $|a|$ and $\|w\|$ so as to guaranteed that that $\dot{V} < 0$. However, it turns out we may identify the essential ones and neglect the remaining terms. This identification is based on the following lemma.

Lemma 1. *The scalar bivariate function*

$$\delta(x,y) = -ax^2 + cx^2 y + dx^3 - ey^4 + fxy^3 + gx^2 y^2 + hx^3 y + kx^4,$$

where a, $c, \cdots, k$ are positive, is locally negative, i.e. there exists an $R > 0$ such that $\delta(x,y) < 0$ for all $0 < x < R$ and $0 < y < R$, provided that $a > 0$ and $e > 0$.

Proof. Rewrite $\delta(x,y)$ as a second degree polynomial in x:

$$
\begin{aligned}
\delta(x,y) &= (-a + cy + dx + gy^2 + hxy + kx^2)x^2 + fy^3 x - ey^4 \\
&=: u(x,y)x^2 + fy^3 x - ey^4.
\end{aligned}
$$

Observe that if

$$x, y < R_1 := \frac{\sqrt{(c+d)^2 + 4a(g+h+k)} - (c+d)}{2(g+h+k)},$$

then

$$
\begin{aligned}
u(x,y) &:= -a + cy + dx + gy^2 + hxy + kx^2 \\
&\leq -a + (c+d)R_1 + (g+h+k)R_1^2 < 0.
\end{aligned}
$$

Hence the leading coefficient of the re-expressed $\delta(x, y)$ is negative. Furthermore, by completing square for $\delta(x, y)$, we obtain

$$\delta(x,y) = u(x,y)[x + \frac{fy^3}{2u(x,y)}]^2 - v(x,y),$$

where

$$v(x,y) = y^4 \frac{-4ae + 4dex + 4cey + (4eg + f^2)y^2 + 4ehxy + 4ekx^2}{4u(x,y)}.$$

Note again that if

$$x, y < R_2 := \frac{\sqrt{(c+d)^2 + 4a(f^2/4e + g + h + k)} - (c+d)}{2(f^2/4e + g + h + k)},$$

then the numerator in $v(x, y)$ is negative. Hence $v(x, y) > 0$ since $u(x, y) < 0$. This yields $\delta(x, y) < 0$ for all $x, y < R := \min(R_1, R_2)$. □

Remark 2. It is easy to show $\delta(x, y)$ may fail to be locally negative by setting $x = 0$ (so that $\delta(0, y) = y^3 - ey^4 > 0$ for any $0 < y < 1/e$) if y^3 presents in $\delta(x, y)$. Furthermore, it is not difficult to see that if the term bx^2y presents in $\delta(x, y)$, then $\delta(x, y)$ remains locally negative as long as $b^2 < 4ae$. In addition, it is also easy to show that $\delta(x, y)$ remains being locally negative if higher order terms are added to $\delta(x, y)$.

Based on Lemma 1, it is easy to show that

Theorem 2 (Case S). The origin of Eq. (1) is locally asymptotically stable provided there exists a cubic form $K(\cdot, \cdot, \cdot)$ such that the following three conditions hold.

(C1) $lQ_0(r, r) = 0,$

(C2) $3K(L_0w, r, r) + 4lQ_0(r, w) + 2w^T ZQ_0(r, r) = 0 \;\forall\; w$

(C3) $2lC_0(r, r, r) + 3K(Q_0(r, r), r, r) < 0.$

Proof. Letting $x := \|w\|$ and $y := |a|$ of $\dot{V}$ in Lemma 4.1, and letting the coefficients of all cubic terms and all quartic terms except a^4 be positive, it is straightforward to show that $-\dot{V}$ is locally positive definite. Note that (C1) implies the absence of y^3-term, (C2) implies $b = 0$, and (C3) is precisely $-e < 0$. □

Clearly, (C1) is a necessary condition. Note that (C1) is precisely the necessary condition of stationary bifurcation $\beta_{1S} = 0$. In addition, (C1) implies that $Q_0(r, r) \in E^s$. since $L_0 : E^s \to E^s$, there exists a unique $w_0 \in E^s$ such that $L_0 w_0 = Q_0(r, r)$. Employing the notion of generalized inverse, we can write w_0 as $L_0^{\dagger} Q_0(r, r)$, where $L_0^{\dagger}$ is the unique inverse of L_0 which satisfies $L_0^{\dagger} L_0 = I|_{E^s}$, the identity map on the subspace E^s. Note that (C2) is a requirement on a function of w, which must be satisfied for all $w \in E^s$. Hence we have, in particular,

$$3K(Q_0(r,r), r, r) + 4lQ_0(r, w_0) + 2w_0^T ZQ_0(r, r) = 0.$$

As a consequence, (C3) can be replaced by (making use of (C1) and (C2))

$$\begin{align} 2lC_0(r,r,r) \quad - & \quad 4lQ_0(r, L_0^{\dagger}Q_0(r,r)) \tag{12} \\ - & \quad Q_0^T(r,r)[L_0^{\dagger^T} Z + ZL_0^{\dagger}]Q_0(r,r) < 0, \tag{13} \end{align}$$

for which there is no need to explicitly solve for the cubic form $K(\cdot,\cdot,\cdot)$.

On the other hand, to finish constructing the Liapunov function $V(x)$ in Eq. (2), $K(x,x,x)$ must be explicitly solved for. By virtue of the algebraic identities of multilinear forms which we derived in [3], $K(x,x,x)$ can be obtained as follows.

Firstly, (C2) is written as

$$[K_{rr}L_0 + 2lQ_{0r} + 2Q_0^T(r,r)Z]w = 0, \ \forall \ w \in E^s.$$

(Here, $K_{rr} := \frac{\partial K(x,r,r)}{\partial x}$ is a row vector. So is lQ_{0r} where $Q_{0r} := \frac{\partial Q_0(x,x)}{\partial x}|_{x=r}$.) Hence the row vector pre-multiplying w must be of the form of kl for some real k. However, although K_{rr} can have any l component, this component is annihilated by L_0 as $lL_0 = 0$. Furthermore, due to (C1), it is easy to show that $Q_0(r,r) \in E^s$ and that lQ_{0r} has no l component. Thus, $Q_0^T(r,r)Z$ also has no l component. Hence k has to be zero. Consequently, we have

$$K_{rr} = -(2lQ_{0r} + 2Q_0^T(r,r)Z)L_0^{\dagger}. \tag{14}$$

As a consequence of (C1) and (C2), (C3) can be expressed as

$$\begin{aligned} & 2lC_0(r,r,r) + K_{rr}Q_0(r,r) \\ & \quad = \quad 2lC_0(r,r,r) - 2lQ_{0r}L_0^{\dagger}Q_0(r,r) - [Q_0(r,r)^T(L_0^{\dagger^T} Z + ZL_0^{\dagger})Q_0(r,r)] \\ & \quad = \quad 2[lC_0(r,r,r) - 2lQ_0(r, L_0^{\dagger}Q_0(r,r)] \\ & \qquad\qquad - [Q_0(r,r)^T(L_0^{\dagger^T} Z + ZL_0^{\dagger})Q_0(r,r)] < 0. \end{aligned}$$

It is clear that Eq. (14) is the only restriction on K. A full parametrization of such K's can be obtained by resorting to the results in [3]. We can of course use the freedom left in choosing K to properly set the coefficients of powers of $|a|$ and $\|w\|$ in $\dot{V}_3$ and $\dot{V}_4$ to zero. (Equivalently, that is to set c, d, f, g, h, k in Lemma 1 to zero. The number of how many of these can be achieved depends on the degree of freedom of K available.)

Since $\lambda_k^{-1} \in \sigma(L_0^{\dagger})$ for $k = 2, ..., n$, clearly, $\Re e\lambda_k^{-1} < 0$. Hence it is not difficult to see that

$$-[Q_0(r,r)^T(L_0^{\dagger^T} Z + ZL_0^{\dagger})Q_0(r,r)]$$

of the equation above contributes adversely to satisfaction of the inequality. However, since we have the freedom in the selection of the c_k's in Eq. (6), as long as they are positive, we may let $c_k \searrow 0$. It is clear that (C3) is equivalent to the

$$2l[C_0(r,r,r) - 2Q_0(r, L_0^{\dagger}Q_0(r,r))] < 0, \tag{15}$$

which is precisely the stability criterion $\beta_{2S} < 0$ for stationary bifurcation.

2.2 Pair of Imaginary Eigenvalues

Since for Case H the eigenvectors r and l are no longer real, the decomposition of x must now be modified accordingly. Explicitly, we must write $x = ar + \bar{a}\bar{r} + w$ and the corresponding new dyadic form becomes

$$P = \bar{l}^T l + l^T \bar{l} + Z. \tag{16}$$

It is straightforward to proceed with the derivation as follows. First we substitute $ar + \bar{a}\bar{w} + w$ into $\dot{V}$. We then separate terms of orders in $|a|$ and $||w||$, and finally we impose appropriate conditions on the relevant coefficients. Detailed will not be given as they are similar to those arising in Case S. We have

Theorem 3 (Case H). The origin of Eq. (1) is locally asymptotically stable stable provided there exists a cubic form $K(\cdot,\cdot,\cdot)$ such that the following seven conditions hold.

(C1) $2\bar{l}Q_0(r,r) + 3i\omega_c K(r,r,r) = 0$,

(C2) $4\bar{l}Q_0(r,\bar{r}) + 2lQ_0(r,r) + 3i\omega K(r,r,\bar{r}) = 0$,

(C3) $4lQ_0(\bar{r},w) + 2w^T ZQ_0(\bar{r},\bar{r}) + 3K(Lw,\bar{r},\bar{r}) - 6i\omega_c K(w,\bar{r},\bar{r}) = 0$,

(C4) $2\bar{l}Q_0(\bar{r},w) + 2lQ_0(r,w) + 2w^T ZQ_0(r,\bar{r}) + 3K(Lw,r,\bar{r}) = 0$, and

(C5) $2\bar{l}C_0(r,r,r) + 3K(Q_0(r,r),r,r) = 0$,

(C6) $3\bar{l}C_0(r,r,\bar{r}) + lC_0(r,r,r) + 3[K(Q_0(r,r),r,\bar{r}) + K(Q_0(r,\bar{r}),r,r)] = 0$,

(C7)
$$\begin{aligned} &2[\bar{l}C_0(r,\bar{r},\bar{r}) + lC_0(r,r,\bar{r})] + 4K(Q_0(r,\bar{r}),r,\bar{r}) \\ &\quad + K(Q_0(\bar{r},\bar{r}),r,r) + K(Q_0(r,r),\bar{r},\bar{r}) < 0 \\ &\quad \Leftrightarrow \Re e[2lC_0(r,r,\bar{r}) + K(Q_0(r,r),\bar{r},\bar{r})] + 2K(Q_0(r,\bar{r}),r,\bar{r}) < 0. \end{aligned}$$

A simple reason for the increase in the number of essential sufficient conditions from 3 for Case S to 7 for Case H can be stated as follows. Firstly, the dimension of the tangent subspace corresponding to the critical eigenvalues is 2 for Case H. Secondly, in contrast to the appearance of real a for Case S, complex numbers now come into play so that each calls for two (independent) conditions for the corresponding coefficient to vanish. In fact, there is no fundamental difficulty in following same approach to study the more degenerate cases such as a double zero at criticality.

Remark 3. Note that in [6], a quartic form must also be included. This is simply because $w \equiv 0$ (in planar case) so that (using the notation in Lemma 1) $\delta(x,y)$ will start, not from

$-ax^2$ but from the cubic terms. Consequently, it is necessary to have b, c, d, f, h all equal to zero, $k < 0$, and $g^2 - 4ek < 0$. These conditions cannot be satisfied without resorting to including a quartic term.

It is crucial to realize that the essential inequality in Case S (C3) and Case H (C7) involves only r but not w. In addition, the remaining equalities in each case enable us to substitute terms of the form $K(Q_0(\cdot,\cdot),\cdot,\cdot)$ by appropriately substituting w in the remaining conditions. This step is exactly what have been produced for Case S (except that $K_{r\bar{r}}$ has to be found besides K_{rr}). In fact, the explicit derivation of (C2) for Case S and (C3)-(C4) for Case H is thus proved indispensable for the subsequent appropriate substitution. (Otherwise, we would have to evaluate the derivative of the Liapunov function only on r (and $\bar{r}$.) For the more degenerate cases, this step can be very complicated.

The equivalence between the formulae for β_{2H}, that is,

$$\beta_{2H} = 2\Re e\{l[2Q_0(r,a) + Q_0(\bar{r},b) + \frac{3}{4}C_0(r,r,\bar{r})]\} \tag{17}$$

where

$$\begin{aligned} a &:= -\frac{1}{2}L_0^{-1}Q_0(r,\bar{r}), && (18)\\ b &:= \frac{1}{2}(2i\omega_c I - L_0)^{-1}Q_0(r,r), && (19) \end{aligned}$$

respectively, and the conditions (C1)-(C7) can be shown in a manner analogous to the procedure which indicates the equivalence between $\beta_{1S} = 0$ and $\beta_{2S} < 0$ and conditions (C1)-(C3). Hence we omit the details.

Remark 4. Allwright and Mees [7] derive a Liapunov function for Hopf bifurcation directly in $\mathbb{R}^n$ and use it to estimate the domain of attraction. Their function consists of quadratic, cubic and quartic terms. Their derivation involves finding a cubic form which cancels out all the cubic terms of the derivative of the Liapunov function along trajectories of the system. The quartic term is then selected to ensure that the resulting derivative is locally negative definite. A primary distinction between our result and that of [7] is that we have taken advantage of the dominancy from the term $w^T(L_0^T P + PL_0)w$ over the stable manifold while they ignored it.

Note that although the apparent discrepancy between β_{2S} and (C3) has been cured by setting $c_k \searrow 0$, a 'smaller' Z reduces the dominancy of $\dot{V}_2$. Hence, a suitable setting of Z, which must take into account the 'distribution' of λ_i, $i = 2, ..., n$ and their inverse, will play a significant role in the robust nonlinear regulation problem.

3 Conclusion and Discussion

In summary, we conclude the following

For Case S The origin of Eq. (1) is locally asymptotically stable provided

1. $lQ_0(r,r) = 0$, and,
2. $2lQ_0(r,w_0) + lC_0(r,r,r) < 0$ where $L_0w_0 = Q_0(r,r)$.

For Case H The origin of Eq. (1) is locally asymptotically stable provided

$$\Re e\{l[2Q_0(r,a) + Q_0(\bar{r},b) + \frac{3}{4}C_0(r,r,\bar{r})]\} < 0.$$

Finally, we claim that it is straightforward to extend this mechanism to constructing Liapunov functions for more degenerate cases such as Case SS (L_0 has eigenvalue zero of multiplicity two and the corresponding Jordan block is diagonal) and Case HH (L_0 has two oscillatory nonresonant modes) simply by modifying P appropriately. In fact, the mechanism hereby presented can be directly adapted to the cases in which the critical eigenvalues of L_0 are all simple. Results for these cases and others (such as S2, defected double zero, not as simle) are in preparation. Applications of the Liapunov functions thus constructed will be further pursued later as well.

Acknowledgement

Research supported in part by the NSF under Grants ECS-86-57561 and CDR-85-00108, and by the AFOSR under AFOSR-87-0073.

References

[1] E.H. Abed and J.-H. Fu, "Local feedback stabilization and bifurcation control, I. Hopf bifurcation," *Systems and Control Letters* Vol. 7 (1986) 11-17.

[2] E.H. Abed and J.-H. Fu, "Local feedback stabilization and bifurcation control, II. Stationary bifurcation," *Systems and Control Letters* Vol. 8 (1987) 467-473.

[3] J.-H. Fu, *Bifurcation Control and Feedback Stabilization*, Ph. D. dissertation, in preparation.

[4] L.N. Howard, Nonlinear Oscillations, in: F.C. Hoppensteadt, Ed., *Nonlinear Oscillations in Biology* (Amer. Math. Society, Providence, RI, 1979) pp. 1-68.

[5] G. Iooss and D.D. Joseph, *Elementary Stability and Bifurcation Theory* Springer-Verlag, New York, 1976.

[6] A.I. Mees and L.O. Chua, "The Hopf bifurcation theorem and its applications to nonlinear oscillations in circuits and systems," *IEEE Trans. Circuits and Systems* CAS-26, 1979, 235-254.

[7] D.J. Allright and A.I. Mees, "Lyapunov method applied to the Hopf bifurcation," *J. London Math. Soc.* (2), 20 (1979), 293-299.

[8] B. Porter, *Modal Control: Theory and Applications*, Taylor and Francis, Ltd., 1979.

[9] T. Poston and I.N. Stewart, *Taylor Expansions and Catastrophes*, Pitman, Research Notes in Math. 7, 1977.

ON STABILIZATION WITH A PRESCRIBED REGION OF ASYMPTOTIC STABILITY

LAHCEN SAYDY,† EYAD H. ABED AND ANDRÉ L. TITS

Department of Electrical Engineering
and the Systems Research Center
University of Maryland, College Park, MD 20742, USA

ABSTRACT

An important unsolved problem in nonlinear control is that of stabilization with a prescribed region of stability. In this paper, sufficient conditions are obtained for the existence of a *linear* feedback stabilizing an equilibrium point of a given nonlinear system with the resulting region of asymptotic stability (RAS) containing a ball of given radius. Conditions for global stabilization are also given. Feedback stabilization is achieved while satisfying a certain robustness property. The technique is applied to planar systems, resulting in a complete design methodology for this case. Examples and simulations illustrating the method are presented.

1. INTRODUCTION

Given a nonlinear control system $\dot{x} = f(x,u)$ where $f : I\!R^n \times I\!R^m \rightarrow I\!R^n$ is a smooth mapping satisfying $f(0,0) = 0$, it is well known that if the associated linearized system is stabilizable, then there exists a linear feedback control which also stabilizes the (null solution of) the original nonlinear system. The null solution of the closed-loop nonlinear system of course possesses a nonempty region of asymptotic stability (RAS). The *size* of the RAS is usually not stated as an *explicit* control objective. The reason for this is the lack of systematic analytical tools for the synthesis of feedback control laws achieving specifications on the RAS. The importance of obtaining such tools is clear, and has been emphasized in [1].

The traditional approach based solely on linearization at an operating point is often considered unreliable from a stability point of view. An alternative design method consists in repeated testing of the performance of the closed-loop system for each of a set of possible stabilizing control laws. Since approximation of the obtained RAS is often very difficult, each of these tests typically involves many simulations of the closed-loop dynamics, and the method is hence very costly [2].

In this paper, sufficient conditions are obtained for the existence of a *linear* feedback stabilizing an equilibrium point of a given nonlinear system with the resulting region of asymptotic stability (RAS) containing a prespecified ball. Conditions for global stabilization are also given. Feedback stabilization is achieved while satisfying a certain robustness property. Synthesis of the desired feedback control laws rests on the solution of certain nonstandard questions in linear systems. These questions are addressed successfully in the case of planar systems, for which a complete design methodology is achieved. Examples and simulations illustrating the method are presented.

---

† On leave from Faculté des Sciences de Marrakech, Université Cadi Ayyad, Marrakech, Morocco.

2. SUFFICIENT CONDITIONS FOR STABILIZATION WITH PRESCRIBED RAS

Before presenting the main results of this section, we establish notation. With S a subset of $\mathbb{C}$, $\Re e(S)$ denotes the set $\{\Re e(s) : \ s \in S\}$. For a real matrix M, $[M]_s$ and $[M]_{ss}$ denote its symmetric and skew-symmetric parts, respectively:

$$[M]_s := \frac{1}{2}(M + M^T),$$

$$[M]_{ss} := \frac{1}{2}(M - M^T).$$

The spectrum of M is denoted by $\sigma(M)$. For x a vector in $\mathbb{R}^n$, $|x|$ denotes its Euclidean norm. Denote by $B(R)$ the open ball in $\mathbb{R}^n$ of radius R centered at the origin. By $\mathbb{R}_-$ and $\mathring{\mathbb{C}}_-$, we intend $(-\infty, 0)$ and the open left-half of the complex plane, respectively. For S a given set, S^n denotes the Cartesian product $S \times S \times \cdots \times S$ (n times).

Two definitions relating to stabilizability of linear systems are now introduced. Let A and B be real matrices of dimensions $n \times n$ and $n \times m$, respectively.

Definition 1. Say that the pair (A,B) is *symmetricallystabilizable*[1] if $\exists K \in \mathbb{R}^{m\times n}$ such that $\sigma([A + BK]_s) \subset \mathbb{R}_-^n$. For Δ a nonempty subset of $\mathbb{R}_-^n$, the pair (A,B) is said to be *symmetrically stabilizable within* Δ if $\forall \Lambda \in \Delta$, $\exists K \in \mathbb{R}^{m\times n}$ such that $\sigma([A + BK]_s) = \Lambda$.

Definition 2. Say that the pair (A,B) is *normally stabilizable* if $\exists K \in \mathbb{R}^{m\times n}$ such that $\sigma(A+BK) \subset \mathring{\mathbb{C}}_-^n$ with $A+BK$ is a normal matrix. Let $\Delta \subset \mathring{\mathbb{C}}_-^n$ be nonempty. Say that (A,B) *is normally stabilizable within* Δ if $\forall \Lambda \in \Delta$, $\exists K \in \mathbb{R}^{m\times n}$ such that $\sigma(A + BK) = \Lambda$ with $A + BK$ normal.

It is a simple exercise to show that if (A,B) is normally stabilizable within Δ then it is symmetrically stabilizablewithin $\Re e(\Delta)$.

We consider nonlinear multi-input control systems $\dot{x} = F(x) + Bu$ where $F : \mathbb{R}^n \to \mathbb{R}^n$ is analytic over $\mathbb{R}^n$ and satisfies $F(0) = 0$, and where B is an $n \times m$ matrix.[2] It is convenient to rewrite the model in the equivalent form

$$\dot{x} = Ax + Bu + h(x), \tag{1}$$

where $A := \frac{\partial F}{\partial x}(0)$ and $h(x)$ represents higher order terms.

Let

$$\mathcal{H} := \{\ h : \mathbb{R}^n \to \mathbb{R}^n, \text{ analytic over } \mathbb{R}^n : h(0) = 0, \ \frac{\partial h}{\partial x}(0) = 0\ \},$$

and define, for a fixed $R > 0$, the following two functions on $\mathcal{H}$:

(i) $\|.\|_R : \mathcal{H} \to \mathbb{R}_+$ is given by

$$\|h\|_R = \sup_{\substack{x\in B(R)\\ x\neq 0}} \frac{|h(x)|}{|x|}$$

---

[1] Recall that Hurwitz stability of $[M]_s$ implies that of M.

[2] Notice that this model is not restrictive since a more general model $\dot{z} = f(z, v)$ may always be put in the form above by letting x be the augmented state $(z,v)^T$ and taking $u = \dot{v}$.

(ii) $\|.\| : \mathcal{H} \to I\!R_+ \cup \{\infty\}$ is given by

$$\|h\| = \sup_{\substack{x \in I\!R^n \\ x \neq 0}} \frac{|h(x)|}{|x|}$$

Notice that $\lim_{|x| \to 0} \frac{|h(x)|}{|x|} = 0$ is well defined and that we allow the possibility $\|h\| = \infty$, with the understanding that inequalities such as $\|h_1 + h_2\| \leq \|h_1\| + \|h_2\|$ are to be interpreted in the obvious way when $\|h_1\|$ or $\|h_2\|$ is infinite. Then $(\mathcal{H}, +, .)$, equipped with either $\|.\|_R$ or $\|.\|$, is a normed vector space. This can be seen easily, noting that analyticity of the elements of $\mathcal{H}$ implies that a function which vanishes within $B(R)$ must indeed vanish everywhere. It is natural to define two balls in $\mathcal{H}$ associated with the norms (i) and (ii):

$$\mathcal{B}_R(\rho) := \{h \in \mathcal{H} : \|h\|_R < \rho\},$$
$$\mathcal{B}(\rho) := \{h \in \mathcal{H} : \|h\| < \rho\}.$$

Theorem 1. Fix $R > 0$ and let (A,B) be symmetrically stabilizablewithin $\Delta \subset (-\infty, -\|h\|_R)^n$ (resp. $(-\infty, -\|h\|)^n$). Then the nonlinear control system (1) is stabilizable within $B(R)$ (resp. globally stabilizable) using linear state feedback.

The proof of this theorem relies on Proposition 1, given next. Let

$$\dot{x} = F(x), \quad F(0) = 0 \tag{2}$$

where F is analytic over $I\!R^n$. Let the null solution of (2) be asymptotically stable in the sense of Lyapunov. Denote the associated RAS by $\mathcal{D}^x$. Consider a change of coordinates $z = Q^T x$ where Q is an *orthogonal* matrix (i.e., Q satisfies $Q^T Q = Q Q^T = I$). In the new coordinates,

$$\dot{z} = \tilde{F}(z) \tag{3}$$

where

$$\tilde{F}(z) := Q^T F(Qz).$$

Clearly, the origin is also asymptotically stable for Eq. (3). The sets $\mathcal{D}^x$ and $\mathcal{D}^z$ are in general different. However, we can use the fact that orthogonal transformations preserve norms and angles to obtain the following proposition.

Proposition 1. The largest Euclidean balls in $\mathcal{D}^x$ and $\mathcal{D}^z$ are identical.

Consequently, for each $R > 0$

$$B(R) \subset \mathcal{D}^x \quad \Longleftrightarrow \quad B(R) \subset \mathcal{D}^z.$$

Proof of Theorem 1: Let $(\lambda_1, \lambda_2, \ldots, \lambda_n) \in \Delta$. Since (A, B) is symmetrically stabilizablewithin Δ, there is a feedback gain matrix K such that

$$\sigma([A + BK]_s) = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}.$$

By setting $u(x) = Kx$ in (1), we obtain the closed loop system

$$\begin{aligned} \dot{x} &= (A + BK)x + h(x) \\ &= [A + BK]_{ss} x + [A + BK]_s x + h(x). \end{aligned} \tag{4}$$

Since $[A+BK]_s$ is symmetric, it can be diagonalized using an orthogonal transformation. Let Q be such a transformation and define new coordinates $z = Q^T x$. Then z satisfies

$$\dot{z} = Gz + Dz + \tilde{h}(z) \tag{5}$$

where

$$D = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n),$$
$$G = Q^T [A+BK]_{ss} Q$$

and

$$\tilde{h}(z) = Q^T h(Qz).$$

Now consider the Lyapunov function candidate $V(z) = \frac{1}{2}|z|^2$ and note that $B(R)$ is a level set for V. The derivative of V along trajectories of (5) is

$$\dot{V}(z) = z^T Gz + z^T Dz + z^T \tilde{h}(z).$$

The first term vanishes since G is skew-symmetric. Thus,

$$\begin{aligned}\dot{V}(z) &\le \sum_{i=1}^{n} \lambda_i z_i^2 + |z|\left|\tilde{h}(z)\right| \\ &\le \max_{1\le i\le n}(\lambda_i)\,|z|^2 + \|h\|_R\,|z|^2 \\ &= \left(\max_{1\le i\le n}(\lambda_i) + \|h\|_R\right)|z|^2 \end{aligned} \tag{6}$$

for all $z \in B(R)$. Noting that $\Delta \subset (-\infty, -\|h\|_R)^n$, we have that $\dot{V}(z) < 0$ for all nonzero $z \in B(R)$. A standard Lyapunov stability result now implies $B(R) \subset \mathcal{D}^z$. In view of Proposition 1, an analogous statement also holds for Eq. (4). This proves the first assertion of Theorem 1. The second assertion (a condition for global asymptotic stability) follows similarly, from the observation that

$$\begin{aligned}\dot{V}(z) &\le \left(\max_{1\le i\le n}(\lambda_i) + \|h\|\right)|z|^2 \\ &< 0\end{aligned}$$

for all $z \in I\!R^n$, $z \neq 0$.

□

Robustness of the stabilization property of Theorem 1 with respect to perturbations in the nonlinear terms $h(x)$ is now considered. The higher order terms $h(x)$ do not affect asymptotic stability of the null solution of a hyperbolic system (linearization with no imaginary eigenvalues). In our framework, we note that for $u(x) = Kx$ a linearly stabilizing feedback, the null solution of $\dot{x} = (A+BK)x + h(x)$ is asymptotically stable for all $h \in \mathcal{H}$. However, the RAS does indeed depend on variations in h. The next result states that the linear feedback $u(x) = Kx$ in Theorem 1 is robust to variations in h. Specifically, the assertion is that $B(R)$ is guaranteed to be within the RAS for each member of a family of systems each of whose linear parts is $\dot{x} = (A+BK)x$.

Theorem 1 (Robustness Form). Let $R > 0$ be fixed. Suppose that (A, B) is symmetrically stabilizable within $\Delta \subset I\!R^n_-$, and let

$$\alpha := \sup_{\Lambda \in \Delta} \max_{1 \leq i \leq n} (\Lambda_i).$$

If $h \in \mathcal{B}_R(|\alpha|)$ (resp. $\mathcal{B}(|\alpha|)$), then the nonlinear control system $\dot{x} = Ax + Bu + h(x)$ is stabilizable within $B(R)$ (resp. globally stabilizable) using linear state feedback.

Under the foregoing assumption, this asserts the existence of a feedback gain matrix $K \in I\!R^{m \times n}$ for which the associated RAS contains $B(R)$, for each $h \in \mathcal{B}_R(|\alpha|)$. No $h \in \mathcal{B}_R(|\alpha|)$ results in an RAS not entirely containing $B(R)$.

3. PLANAR SYSTEMS

In this section, we apply the results of Section 2 to planar systems, i.e., systems as in (1) with $A \in I\!R^{2 \times 2}$ and $b \in I\!R^2 \setminus \{0\}$.[3] Let

$$\Delta := \{ (\lambda_1, \lambda_2) \in I\!R_- \times I\!R_- : \ \lambda_1 \neq \lambda_2 \text{ and } \lambda_1, \lambda_2 \notin \sigma(A) \},$$

$$\nu(A, b) := \frac{b^T \operatorname{Adj}(A) b}{|b|^2} \quad \text{and} \quad \mu(A, b) := \frac{|\operatorname{Adj}(A) b|^2}{|b|^2}.$$

Also, define the set $\Delta(A,b)$ by

$$\Delta(A, b) = \{(\lambda_1, \lambda_2) \in \Delta : \ \lambda_1 \lambda_2 - \nu(A, b)(\lambda_1 + \lambda_2) + \mu(A, b) = 0\}, \tag{7}$$

where $\operatorname{Adj}(A)$ denotes the adjugate of A. The defining equation in (7) is that of a hyperbola. This hyperbola is equivalently characterized by

$$\lambda_2 = \frac{\nu(A, b)\lambda_1 - \mu(A, b)}{\lambda_1 - \nu(A, b)}. \tag{8}$$

The following theorem gives a necessary and sufficient condition for (A, b) to be normally stabilizable within $\Delta(A, b)$.

Theorem 2. The pair (A, b) is normally stabilizable within $\Delta(A, b)$ if and only if $\nu(A, b) < 0$. Furthermore, given any set of desired closed-loop eigenvalues $(\lambda_1, \lambda_2) \in \Delta(A, b)$, the corresponding normally stabilizing feedback gain is given by

$$k = [1 \ \ 1] \, [\, (\lambda_1 I - A)^{-1} b \ \ (\lambda_2 I - A)^{-1} b \,]^{-1}. \tag{9}$$

Proof of Theorem 2: First, we show that $\Delta(A, b) \neq \emptyset$. From Eq. (8), we obtain

$$\frac{\partial \lambda_2}{\partial \lambda_1} = \frac{\mu(A, b) - \nu^2(A, b)}{(\lambda_1 - \nu(A, b))^2}. \tag{10}$$

---

[3] If we let B be a nonzero 2×2 matrix, then it is either nonsingular, in which case the stabilization problem becomes trivial, or of rank one. The latter case is equivalent to considering B to be a vector b in $I\!R^2$.

Defining C to be the matrix $[b \;\; \mathrm{Adj}(A)b]$, and noting that

$$\det(C^T C) = (\det(C))^2 = |b|^4 \left(\mu(A,b) - \nu^2(A,b)\right), \tag{11}$$

we determine that λ_2 is a monotonically increasing function of λ_1. A quick sketch of the hyperbola (8) (i.e., a plot of λ_2 vs. λ_1) convinces us that $\Delta(A,b) \cap I\!R^2_- \neq \emptyset$ precisely when $\nu(A,b) < 0$. The sketch just referred to is also useful in finding pairs $(\lambda_1, \lambda_2) \in \Delta(A,b)$. To find such a pair, we may simply pick a value λ_1 in $(\nu(A,b), 0) \setminus \sigma(A)$, and then use Eq. (8) to compute the corresponding value of λ_2.

Next, we show that for $(\lambda_1, \lambda_2) \in \Delta(A,b)$, the vectors

$$v_i = (\lambda_i I - A)^{-1} b \,, \qquad i = 1,2$$

are orthogonal. It is equivalent to show that the vectors w_1 and w_2 are orthogonal, where

$$w_i = \chi_A(\lambda_i)\; v_i \,, \qquad i = 1,2$$

and $\chi_A(s)$ denotes the characteristic polynomial of A. (Recall that $\lambda_1,\ \lambda_2 \notin \sigma(A)$.) We have

$$\begin{aligned} w_1^T w_2 &= b^T \left(\mathrm{Adj}(\lambda_1 I - A)\right)^T \mathrm{Adj}(\lambda_2 I - A) b \\ &= b^T \left(\lambda_1 \lambda_2 I - \lambda_1 \mathrm{Adj}(A) - \lambda_2 \mathrm{Adj}^T(A) + \mathrm{Adj}^T(A)\mathrm{Adj}(A)\right) b \\ &= |b|^2 \lambda_1 \lambda_2 - (\lambda_1 + \lambda_2) b^T \mathrm{Adj}(A) b + |\mathrm{Adj}(A) b|^2 . \end{aligned}$$

Since $(\lambda_1, \lambda_2) \in \Delta(A,b)$, it follows that $w_1^T w_2 = 0$.

We now show that v_1 and v_2 are eigenvectors of $A + bk$ corresponding to λ_1 and λ_2, respectively. Let $V = [v_1 \;\; v_2]$. Then $k = [1 \;\; 1]V^{-1}$ and

$$\begin{aligned} [\,(A+bk)v_1 \;\; (A+bk)v_2\,] &= (A+bk)V = AV + bkV \\ &= [Av_1 + b \;\; Av_2 + b]. \end{aligned}$$

On the other hand, $Av_i + b = \left(A(\lambda_i I - A)^{-1} + I\right) b = \lambda_i v_i$ for $i = 1,2$. Therefore

$$(A + bk)v_i = \lambda_i v_i, \qquad i = 1,2,$$

i.e., $\sigma(A+bk) = \{\lambda_1, \lambda_2\}$. Since the eigenvectors v_1 and v_2 are orthogonal, we obtain that $A + bk$ is a normal matrix (in fact symmetric since λ_1 and λ_2 are real).

□

Remarks:

1) The condition $\nu(A,b) < 0$ implies that $\det[b \;\; \mathrm{Adj}(A)b] \neq 0$, hence (A,b) controllable since $\det[b \;\; Ab] = -\det[A \;\; \mathrm{Adj}(A)b]$, in all but the case when b and $\mathrm{Adj}(A)b$ are of opposite directions. Also, if (A,b) is not controllable, then Eqs. (10)-(11) imply that $\frac{\partial \lambda_2}{\partial \lambda_1} = 0$, hence Eq. (8) corresponds to the horizontal line $\lambda_2 = \nu$.
2) Note that it is not possible to force *both* eigenvalues to be arbitrarily large. Clearly, this is a consequence of the normality requirement. It can be shown that in order for a pair (A,B) to be "arbitrarily" normally stabilizable (normally stabilizable with arbitrarily negative assignable closed-loop eigenvalues) it is necessary and sufficient that rank $(B) = n$, n being the size of A.

The following theorem is a direct consequence of Theorems 1 and 2. Fix $R > 0$.

Theorem 3. Assume that $\nu(A,b) < 0$. If $h \in \mathcal{B}_R(|\nu(A,b)|)$ (resp. $\mathcal{B}(|\nu(A,b)|)$) then there exists a linear feedback $u(x) = kx$ such that (the origin of) the closed-loop system $\dot{x} = (A + bk)x + h(x)$ is asymptotically stable within $B(R)$ (resp. globally asymptotically stable). Furthermore, for any desired closed-loop eigenvalues $\lambda_1 \in (\ \nu(A,b), -\|h\|_R\ ) \setminus \sigma(A)$ (resp. $(\nu(A,b), -\|h\|\ ) \setminus \sigma(A)$) and λ_2 given by (8), the feedback gain k is given by (9).

The analogue of Theorem 1 (Robustness Form) in the two-dimensional case follows by taking α to be any number strictly between $\nu(A,b)$ and 0. The desired feedback k is obtained in a manner identical to that outlined in Theorem 3, with $-\|h\|_R$ (resp. $-\ \|h\|$) replaced by α.

4. EXAMPLES

Example 1. Let $R = 1$ and consider the system

$$\dot{x}_1 = -\frac{3}{2}x_1 + x_2 + x_1^2$$
$$\dot{x}_2 = x_2 + u - x_2^2.$$

The origin of the unforced system is unstable since $\sigma(A) = \{-\frac{3}{2}, 1\}$. It is easily checked that (A,b) is controllable, $\nu(A,b) = -\frac{3}{2}$, $\|h\|_{R=1} = 1 < |\nu(A,b)|$ and $\mu(A,b) = \frac{13}{4}$. By picking λ_1 in $(-\frac{3}{2}, -\|h\|_R\ ) \setminus \sigma(A) = (-\frac{3}{2}, -1)$, say $\lambda_1 = -\frac{5}{4}$, we get from (8) that $\lambda_2 = -\frac{11}{2}$ and from (9) that $k = [1 \quad -\frac{25}{4}]$. The closed-loop system is

$$\dot{x}_1 = -\frac{3}{2}x_1 + x_2 + x_1^2$$
$$\dot{x}_2 = x_1 - \frac{21}{4}x_2 - x_2^2.$$

Simulations of the closed-loop system for various initial conditions, shown in Fig. 1, corroborate the fact that $B(1)$ is contained in the actual RAS. Note that some initial conditions in the immediate vicinity of $B(1)$ lead to instability (e.g., $x_0 = (1.4, 0)$ and $(1.2, 1.2)$).

Example 2. We consider globally stabilizing the system

$$\dot{x}_1 = -\frac{3}{2}x_1 + x_2$$
$$\dot{x}_2 = -x_1 + x_2 + u + \sin(x_1).$$

Here, A and b are as in Example 1, $h(x) = [0, \sin(x_1) - x_1]^T$ and $\|h\| = 1.217$. Proceeding as in Example 1, we obtain by choosing $\lambda_1 = -\frac{5}{4}$ that the closed-loop system is

$$\dot{x}_1 = -\frac{3}{2}x_1 + x_2$$
$$\dot{x}_2 = -\frac{21}{4}x_2 + \sin(x_1).$$

With the Lyapunov function $V(x) = x_1^2 + x_2^2$, we find that $\dot{V}(x) < -x_1^2 - \frac{17}{2}x_2^2 < 0$ for all $x \neq 0$. Therefore, the null solution of the closed-loop system is globally asymptotically stable, as predicted by Theorem 3.

5. CONCLUSION

Sufficient conditions for stabilizability of nonlinear systems with a region of asymptotic stability containing a prescribed ball in $I\!R^n$ have been presented. Under a symmetric stabilizability condition on the system linearization, it was shown that there is a linear stabilizing controller, and that the closed-loop system stability is robust to certain model perturbations. Necessary and sufficient conditions for normal stabilizability of a two-dimensional linear time-invariant system were obtained. These facilitated identification of a closed-form formula for a stabilizing feedback gain k which guarantees stabilization within a given ball. Necessary and sufficient conditions for normal stabilizability of a general pair (A,B) is an issue that is currently being considered.

Acknowledgment. The authors' research is supported in part by the US National Science Foundation under Grants ECS-86-57561, DMC-84-51515 and CDR-85-00108, and by Air Force Office of Scientific Research Grant AFOSR-87-0073.

REFERENCES

[1] Many Authors, "Challenges to control: A collective view," *IEEE Trans. Automatic Control*, Vol. 32, 1987, pp. 275-285.

[2] E. Polak and D.Q. Mayne, "Design of nonlinear feedback controllers," *IEEE Trans. Automatic Control*, Vol. 26, 1981, pp. 730-733.

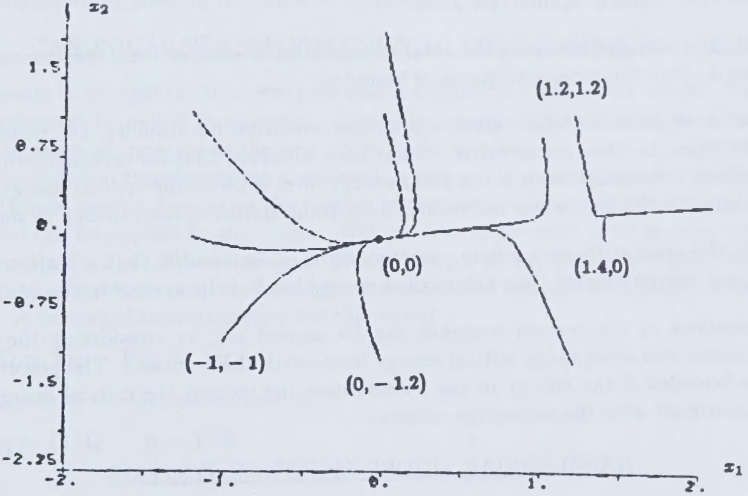


Fig. 1 Closed loop trajectories for Example 1

APPLICATION OF HAMILTONIAN SYSTEM THEORY TO THREE-MACHINE POWER SYSTEM TRANSIENT ANALYSIS

Tao Guo
Dept. of Elec. Eng. &
Computer Science
University of Michigan
Ann Arbor, MI 48109

Natarajan Narasimhamurthi
Dept. of Elec. Eng. &
Computer Science
Univ. of Michigan-Dearborn
Dearborn, MI 48128

N. Harris McClamroch
Dept. of Aerospace Eng.
University of Michigan
Ann Arbor, MI 48109

ABSTRACT

Hamiltonian system theory is applied to analyze the transient response of three-machine power systems. By studying the energy in the cutsets and the geometric property of KAM surfaces, it is shown that a trajectory of a three-machine power system may be bounded even if the corresponding energy level exceeds the critical energy level of the system, as defined in the existing literature.

INTRODUCTION

In the transient analysis of power systems, the approach based on energy integrals is as follows:

1. One first constructs an energy integral which is constant along any trajectory.
2. One next computes the energy level at each unstable equilibrium point near the stable equilibrium point; the smallest of these levels is often referred to as the *critical energy level of the system* and the corresponding equilibrium point as the *critical unstable equilibrium point.*
3. If after any disturbance, the total energy level is smaller than the critical energy level, then the system response is bounded.

The above method, which gives a sufficient condition for stability, i.e. boundedness of trajectories, is often conservative. It has been observed that for certain disturbances, a trajectory is bounded even if the total energy level exceeds the critical energy level of the system. In this paper, we use results from Hamiltonian system theory to show that

1. In the case of three-machine power systems, it is possible that a trajectory with total energy greater than the critical energy level of the system is bounded.
2. Analysis of the system response can be carried out by considering the various cutsets and associating critical energy levels with these cutsets. The power system is bounded if the energy in the cutsets does not exceed the critical energy levels associated with the respective cutsets.

HAMILTONIAN MODEL OF POWER SYSTEMS

The Hamiltonian model of a power system with damping ignored is [4] [5]

$$\dot{\mathrm{q}} = \mathrm{M}^{-1}\mathrm{p} \tag{1}$$

$$\dot{\mathrm{p}} = \mathrm{P}^{m} - \mathrm{P}^{e}(\mathrm{q}) \tag{2}$$

Here $\mathbf{q} = \boldsymbol{\delta} - \boldsymbol{\delta}^s$ is the *position* vector and $\mathbf{p} = \mathbf{M}\boldsymbol{\omega}$ is the *momentum* vector; $\boldsymbol{\delta}$ and $\boldsymbol{\omega}$ are the vectors of power angles of the machines and their angular speeds; $\mathbf{M}$ is a constant diagonal matrix whose positive entries are the inertias of the machines. $\mathbf{P}^m$ is a vector whose entries are the constant mechanical power input to the machines. $\mathbf{P}^e$ is defined through

$$\mathbf{P}^e(\mathbf{q}) = \begin{bmatrix} \sum_{j=1}^{n} F_{1j}\sin(q_{1,j} + \delta^s_{1,j}) \\ \vdots \\ \sum_{j=1}^{n} F_{nj}\sin(q_{n,j} + \delta^s_{n,j}) \end{bmatrix} \tag{3}$$

which is the vector of electrical power delivered to the system through each machine, and $q_{i,j} = q_i - q_j$ and $\delta^s_{i,j} = \delta^s_i - \delta^s_j$. The constant $\boldsymbol{\delta}^s$ satisfies $\mathbf{P}^m - \mathbf{P}^e(\mathbf{0}) = \mathbf{0}$.

The Hamiltonian function, which also serves as an energy function, is

$$H(\mathbf{q}, \mathbf{p}) = KE(\mathbf{p}) + PE(\mathbf{q}) \tag{4}$$

where

$$\begin{aligned} KE(\mathbf{p}) &= \frac{1}{2}\mathbf{p}^T\mathbf{M}^{-1}\mathbf{p} && (5) \\ PE(\mathbf{q}) &= -\mathbf{q}^T\mathbf{P}^m - FC(\mathbf{q}) && (6) \end{aligned}$$

are the *kinetic energy* and the *potential energy* respectively, and

$$\begin{aligned} FC(\mathbf{q}) &= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} F_{ij}[\cos(q_{i,j} + \delta^s_{i,j}) - \cos(\delta^s_{i,j})] \\ &= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\int_0^{q_i} F_{ij}\sin(\xi_{i,j} + \delta^s_{i,j})d\xi_i && (7) \end{aligned}$$

The vector $(\mathbf{q}, \mathbf{p}) = (\mathbf{0}, \mathbf{0})$ is a local minimum of the Hamiltonian function so that it is a stable equilibrium point in the sense of Lyapunov.

CANONICAL TRANSFORMATION TO CUTSET COORDINATES

In order to analyze the transient response of a three-machine power system, the system is transformed to cutset coordinates. First, the canonical transformation is presented; then the physical meaning of the new coordinates is given.

The canonical transformation developed in this paper is based on the knowledge of a cutset whose motion dominates the motion of the system. The canonical transformation obtained can be applied to any cutset. *Without loss of generality, we assume the motion of the cutset* $(1,2)$-$(1,3)$ *is dominant.* This cutset partitions the system into two subsystems: one subsystem consists of machine 1 and another subsystem consists of machine 2 and 3. A canonical transformation for this cutset is

$$\mathbf{Q} = \mathbf{T}\mathbf{q} \qquad \mathbf{P} = \mathbf{T}^{-T}\mathbf{p} \tag{8}$$

The inverse of the transformation is

$$\mathbf{q} = \mathbf{T}^{-1}\mathbf{Q} \qquad \mathbf{p} = \mathbf{T}^T\mathbf{P} \tag{9}$$

where

$$\begin{aligned} \mathbf{q}^T &= [q_1 \;\; q_2 \;\; q_3] \qquad \mathbf{p}^T = [p_1 \;\; p_2 \;\; p_3] && (10) \\ \mathbf{Q}^T &= [Q_1 \;\; Q_2 \;\; Q_3] \qquad \mathbf{P}^T = [P_1 \;\; P_2 \;\; P_3] && (11) \end{aligned}$$

and

$$\mathbf{T} = \begin{bmatrix} 1 & -M_2/M_{23} & -M_3/M_{23} \\ 0 & 1 & -1 \\ M_1/M & M_2/M & M_3/M \end{bmatrix}, \tag{12}$$

here $M = \sum_{i=1}^{3} M_i$ and $M_{23} = M_2 + M_3$. Then the Hamiltonian function of the system in the new coordinates is

$$H(\mathbf{Q}, \mathbf{P}) = \frac{1}{2}\mathbf{P}^T \hat{\mathbf{M}}^{-1}\mathbf{P} + \hat{PE}(\mathbf{Q}) \tag{13}$$

where $\hat{PE}(\mathbf{Q}) = PE(\mathbf{q})$ and

$$\hat{\mathbf{M}}^{-1} = \text{diag}\,[1/M_1 + 1/M_{23}, \;\; 1/M_2 + 1/M_3, \;\; 1/M]. \tag{14}$$

Remarks:

1. q_1 and $(M_2q_2 + M_3q_3)/M_{23}$ are the positions of the center of gravities of the two subsystems connected by the cutset (1, 2)-(1, 3). Then $Q_1 = q_1 - (M_2q_2 + M_3q_3)/M_{23}$ is the difference of these two positions.

2. $Q_2 = q_2 - q_3$ represents the difference of the positions inside the subsystem consisting of machines 2 and 3.

3. $Q_3 = (M_1q_1 + M_2q_2 + M_3q_3)/M$ is the position of the center of gravity of the original system, which, without loss of generality, can be assumed to be zero. Under this assumption, the Hamiltonian function becomes

$$H(Q_1, Q_2, 0, P_1, P_2, P_3) = \frac{1}{2}\mathbf{P}^T \hat{\mathbf{M}}^{-1}\mathbf{P} + \hat{PE}(Q_1, Q_2, 0) \tag{15}$$

Then we have $\dot{P}_3 = -\partial H/\partial Q_3 = 0$ i.e., $P_3 = c$ where c is a constant. Therefore the system can be reduced to two degrees of freedom. For simplicity, the variables (Q_3, P_3) will be dropped from $(\mathbf{Q}, \mathbf{P})$ in the rest of the paper.

The Hamiltonian function is readily decomposed into the sum of a Hamiltonian function of an integrable system and a perturbation:

$$H(Q_1, Q_2, P_1, P_2) = H_0(Q_1, P_1, Q_2, P_2) + H_1(Q_1, Q_2) \tag{16}$$

where

$$H_0(Q_1, P_1, Q_2, P_2) = H_0^c(Q_1, P_1) + H_0^p(Q_2, P_2) \tag{17}$$

is the Hamiltonian function of an integrable system(see Appendix A);

$$H_0^c(Q_1, P_1) \;\; = \;\; \frac{1}{2}P_1^2(\frac{1}{M_1} + \frac{1}{M_{23}}) + \hat{PE}(Q_1, 0) \tag{18}$$

$$H_0^p(Q_2, P_2) \;\; = \;\; \frac{1}{2}P_2^2(\frac{1}{M_2} + \frac{1}{M_3}) + \hat{PE}(0, Q_2); \tag{19}$$

the former is a Hamiltonian function for the cutset, and the latter is a Hamiltonian function for the subsystem consisting of machines 2 and 3. Our assumption that the motion of the cutset (1, 2)-(1, 3) is dominant corresponds to the assumption that $H_0^c(Q_1, P_1) \gg H_0^p(Q_2, P_2)$. $H_1(Q_1, Q_2) = \hat{PE}(Q_1, Q_2) - \hat{PE}(Q_1, 0) - \hat{PE}(0, Q_2)$ is the perturbation.

ANALYSIS OF THREE-MACHINE POWER SYSTEM TRANSIENT RESPONSE

This section analyzes the transient response of three-machine power systems by studying the energy in cutsets of this class of power systems and the geometric property of the KAM(Kolmogorov-Arnold-Moser) surfaces of two-degree of freedom Hamiltonian systems(see Appendix A).

First, we study the unperturbed system defined by (17). In this case, the system consists of two independent subsystems defined by $H_0^c(Q_1, P_1)$ and $H_0^p(Q_2, P_2)$ respectively. Since there is no coupling between the subsystems, there is no energy transfer between the subsystems. Thus $H_0^c(Q_1, P_1)$ and $H_0^p(Q_2, P_2)$ are two first integrals of the unperturbed system, i.e., $H_0^c(Q_1, P_1) = h^c$ and $H_0^p(Q_2, P_2) = h^p$ where h^c and h^p are two constants. Then each subsystem can be studied independently in the two-dimensional phase space (Q_i, P_i) for $i = 1, 2$.

The behavior of each subsystem is the same as that of a pendulum with constant input. There are two equilibrium points in each subsystem: a stable one, (Q_i^s, P_i^s), and an unstable one, (Q_i^u, P_i^u) for $i = 1, 2$. The energy level of a subsystem at its unstable equilibrium point is called the *critical energy level of the subsystem.* The critical energy level of the subsystem $H_0^c(Q_1, P_1)$ is called the *critical energy level of the cutset.* It is important to note that the critical energy level of a cutset may exceed the critical energy level of the original system. The trajectory of a subsystem corresponding to the critical energy level of a subsystem defines a separatrix of the subsystem. If the energy level of a subsystem is smaller than its critical energy level, trajectories of the subsystem are closed curves around the stable equilibrium point of the subsystem in the two-dimensional phase space. If the energy level of a subsystem is greater than its critical energy level, trajectories of the subsystem are not bounded.

Now consider the four-dimensional unperturbed system (17). The equilibrium points of the unperturbed system are simple combinations of the equilibrium points of the subsystems. There is a stable equilibrium point of the unperturbed system:$(Q_1^s, Q_2^s, P_1^s, P_2^s)$, and there are three unstable equilibrium points:$(Q_1^u, Q_2^u, P_1^s, P_2^s)$, $(Q_1^s, Q_2^s, P_1^u, P_2^u)$ and $(Q_1^u, Q_2^u, P_1^u, P_2^u)$. For a given energy level h, $H_0(Q_1, Q_2, P_1, P_2) = h$ defines a three-dimensional iso-energy surface in the four-dimensional phase space (Q_1, Q_2, P_1, P_2). Trajectories of the system are restricted to this iso-energy surface. Also, a trajectory of the unperturbed system is a cross product of trajectories of the subsystems with $(Q_1(t), P_1(t), Q_2(t), P_2(t)) \in \mathcal{R}^2 \times \mathcal{R}^2$, $t \geq 0$. Therefore it is not difficult to prove that the unperturbed system is bounded if and only if the energy level of each subsystem is smaller than its critical energy level.

To examine the behavior of the unperturbed system further, we assume that the unperturbed system is transformed to *action-angle* coordinates by a canonical transformation as given in [3, section 1.2 and 1.3]:

$$H_0(I_1, I_2, \theta_1, \theta_2) = H_0^c(I_1) + H_0^p(I_2) \tag{20}$$

where I_1 and I_2 are the action variables, and θ_1 and θ_2 are the angle variables. Since the energy levels of the subsystems are constant along any trajectory of the unperturbed system: $H_0^c(I_1) = H_0^c(Q_1, P_1) = h^c$ and $H_0^p(I_2) = H_0^p(Q_2, P_2) = h^p$, the actions I_1 and I_2 are constant. Also, the angular velocities

$$\dot{\theta}_1 = \frac{\partial H_0^c(I_1)}{\partial I_1} \qquad \dot{\theta}_2 = \frac{\partial H_0^p(I_2)}{\partial I_2} \tag{21}$$

are constant. The family of KAM surfaces of the unperturbed system can be illustrated as follows. For energy levels of the subsystems smaller than their critical energy levels, the unperturbed system can be viewed as a system consisting of two hinged rotating rods(see Fig. 1). The subsystem $H_0^c(I_1)$ can be viewed as rod 1: the length of the rod is I_1 and one end of the rod is fixed; the rod rotates horizontally around the fixed end with constant angular velocity $\dot{\theta}_1$. The subsystem $H_0^p(I_2)$ can be viewed as rod 2: the length of the rod is I_2 and one end is hinged to the moving end of rod 1; the rod rotates vertically around the hinged end with constant angular velocity $\dot{\theta}_2$. The path of the moving end of rod 2 represents a trajectory of the unperturbed system, and the path is located on a two-dimensional torus–a KAM surface–in three dimensional space. It is obvious that KAM surfaces exist only in regions where the energy levels of the subsystems are smaller than their critical energy levels, for which the trajectories of the subsystems are closed curves.

Next, we study the perturbed system (16). The behavior of the perturbed system may be topologically different from the unperturbed system. The unstable equilibrium points and the critical energy levels of the subsystems may be slightly changed [1], and chaotic motion can occur in the perturbed system, especially in the neighborhood of the separatrices of the subsystems [7][1][2][3]. In any region of chaotic motion the KAM surfaces are destroyed. However, the KAM theorem guarantees that, for a sufficiently small perturbation, most of the KAM surfaces are preserved but deformed.

In action-angle coordinates, the perturbed system is

$$H(I_1, I_2, \theta_1, \theta_2) = H_0^c(I_1) + H_0^p(I_2) + H_1(I_1, I_2, \theta_1, \theta_2) \tag{22}$$

Due to the perturbation, the functions $H_0^c(I_1)$ and $H_0^p(I_2)$ are not constant, and there is energy transfer between the subsystems. If an initial state of the perturbed system is located on a KAM surface, the motion of the perturbed system is near-periodic and bounded; the variations of I_1 and I_2 and the energy transfer between the subsystems are near-periodic and small. If an initial state of the perturbed system is not located on a KAM surface but is enclosed by a KAM surface, the motion is still bounded. Therefore, for any initial energy levels of the subsystems smaller than the critical energy levels of the subsystems, the motion of the perturbed system is bounded if the perturbation is sufficiently small.

Thus, summarizing the above argument, a conclusion can be drawn about the determination of the stability of a three-machine power system: if the initial energy level of the dominant cutset is smaller than its critical energy level, then the trajectory is bounded if the perturbation is sufficiently small. In view of the fact that the motion of a power system is dominated by a cutset, the critical energy level of the cutset dominating the system quantitatively determines the stability of the system. A number of papers [6, and references listed therein] have discussed methods for determining the unstable equilibrium points associated with cutsets.

NUMERICAL STUDY OF A THREE-MACHINE POWER SYSTEM

In what follows, a three-machine power system example is studied. We show that KAM surfaces are observed at energy levels greater than the critical energy level of the system. This observation implies that a three-machine power system may be bounded even if the total energy of the system exceeds the critical energy level of the system.

---

[1] Here, we ignore possible bifurcation phenomena in which the number of unstable equilibrium points of the perturbed system are different from that of the unperturbed system.

The parameters and the structure of the system are given in Appendix B. The cases of three-bus ground faults at machine 1 and 2 are studied for different fault durations. The canonical transformation is applied to the system according to the cutset dominating the system. Along with the digital simulations, we record the image of the Poincare map(see Appendix A) with the surface of section defined by $(Q_2 = 0, P_2 \geq 0)$.

Three-bus ground fault at machine 1: The fault at machine 1 causes the system to swing at cutset $(1,2)$-$(1,3)$; the associated unstable equilibrium point is $(2.517, 0.574, -0.107)$ whose energy level, which is both the critical energy level of the cutset and the critical energy level of the system, is 14.941.

1. *The energy level of the system is smaller than the critical energy level of the cutset:* In this case, the fault is cleared after 0.8 seconds; the energy level of the system is 13.398 which is smaller than the critical energy level of the cutset, 14.941. The transient of the system after the clearance of fault is simulated; the image of the Poincare map with the surface of section defined by $(Q_2 = 0, P_2 \geq 0)((q_{2,3} = 0, p_{2,3} \geq 0))$ is shown in Figure 3. The intersections of the trajectory with the surface of section form a closed curve corresponding to a KAM surface; hence the motion is bounded.

2. *The energy level of the system is greater than the critical energy of the cutset:* This case is the same as the above except that the fault is cleared after 0.839 seconds; the energy level of the system is 15.087 which is slightly higher than the critical energy level of the cutset, 14.941. The trajectory intersects the surface of section a few times, and, after 98.3 seconds, blows up.

Three-bus ground fault at machine 2: The fault at machine 2 causes the system to swing at cutset $(2,1)$-$(2,3)$; the associated unstable equilibrium point is $(0.368, 2.797, -0.221)$ whose energy level, which is the critical energy level of the cutset, is 17.420 which is greater than the critical energy level of the system, 14.941.

1. *The energy level of the system is smaller than the critical energy level of the cutset but greater than the critical energy level of the system:* In this case, the fault is cleared after 1.22 seconds; the energy level of the system is 16.387 which is smaller than the critical energy level of the cutset, 17.420, but greater than the critical energy level of the system, 14.941. The transient of the system after the clearance of fault is simulated; the image of the Poincare map with the surface of section defined by $(Q_2 = 0, P_2 \geq 0)$ $((q_{1,3} = 0, p_{1,3} \geq 0))$ is shown in Figure 4. The intersections of the trajectory with the surface of section form a closed curve corresponding to a KAM surface; hence the motion is bounded.

2. *The energy level of the system is greater than the critical energy of the cutset:* This case is the same as the above except that the fault is cleared after 1.266 seconds; the energy level of the system is 17.818 which is slightly greater than the critical energy level of the cutset, 17.420. The trajectory intersects the surface of section a few times, and, after 17.0 seconds, blows up.

CONCLUSION

The results of this study show that the transient stability of a three-machine power system is determined by the critical energy levels of individual cutsets rather than the critical energy level of the system. Among all the cutsets is a cutset corresponding to the greatest energy level, thus dominating the motion of the system. Also the results suggest that the transient response of a three-machine power system can be determined by distinguishing the dominant cutset based on knowledge of the initial state of the system and by comparing the initial energy level of the system with the critical energy level of the cutset: if the initial energy level of the system is smaller than this critical energy level, the system is bounded.

The results obtained for three-machine power systems can not be applied to power systems with more than three machines, since n-dimensional KAM surfaces, $n > 2$, no longer separate the $(2n-1)$-dimensional iso-energy surface into isolated regions. In Hamiltonian systems with more than two degrees of freedom, for almost every initial condition the trajectory explores almost every part of the iso-energy surface. This phenomenon is called Arnold diffusion. As an extension of the research presented in this paper, a different approach has been suggested for the transient analysis of power systems with more than three machines. The results will be presented elsewhere.

APPENDIX A: BRIEF REVIEW OF KAM SURFACES AND THEIR GEOMETRIC PROPERTIES

A well studied problem in Hamiltonian system theory is the nature of solutions on an iso-energy surface. An iso-energy surface is a $(2n-1)$-dimensional manifold defined by $H(\mathrm{q}, \mathrm{p}) = h$, where h is a constant and n is the number of degrees of freedom of the system. Trajectories of the system are located on the iso-energy surface. Suppose that another function F is also constant along any solution of the system. Then the iso-F surface will also give rise to a $(2n-1)$-dimensional manifold which further restricts the motion of the system. If this surface intersects the iso-energy surface transversally, the resulting intersection of the iso-F surface and the iso-energy surface is a $(2n-2)$-dimensional manifold. In Hamiltonian system theory, such functions that are constant along every trajectory are called *first integrals*, and if the manifolds defined by first integrals intersect transversally, they are said to be independent. Suppose there are n independent first integrals; then the intersection of the manifolds will result in an n-dimensional manifold which is called a KAM surface. A KAM surface is topologically equivalent to an n-dimensional torus on which trajectories of the system wind. Hamiltonian systems for which there exist n independent first integrals are called integrable Hamiltonian systems.

However, most dynamic systems are not integrable. The behavior of non-integrable systems may be topologically different from that of integrable systems. It has been shown that non-integrable systems may give rise to chaotic motion in the sense that it is impossible to predict the behavior of a trajectory based on knowledge of a nearby trajectory[7] [1][3]. It can also be shown that every non-linear Hamiltonian system is approximately integrable in the sense that the Hamiltonian function can be represented by an integrable Hamiltonian function plus a perturbation. The KAM theorem guarantees that if the perturbation is sufficiently small, most of the KAM surfaces of the integrable system are preserved but deformed by the perturbation [3].

The most convenient way to examine the KAM surfaces is by a *Poincare map*. The Poincare map represents the sequential intersections of a trajectory of the system with a surface of section [7] [3]. One way to define a $(2n-2)$-dimensional surface of section is shown in the following: since the Hamiltonian function is constant along each trajectory of the system, $H(\mathbf{q},\mathbf{p}) = h$, then one component of $\mathbf{p}$, say p_n, can be expressed as

$$p_n = p_n(q_1, \cdots, q_n, p_1, \cdots, p_{n-1}, h). \tag{23}$$

Then a $(2n-2)$-dimensional surface of section is defined by the conditions that p_n is non-negative and q_n is a constant, usually zero.

For instance, let a surface of section in a two-degree of freedom Hamiltonian system be defined by $(q_2 = 0, p_2 \geq 0)$. Then the image of the Poincare map is defined by the intersections of a trajectory with the plane (q_1, p_1). If a trajectory lies on a KAM surface, the intersections of the trajectory with the plane (q_1, p_1) form a dense set of a closed curve in the plane (q_1, p_1).

There is a well-known geometric property of KAM surfaces in a two-degree of freedom Hamiltonian system: two-dimensional KAM surfaces are equivalent to two-dimensional tori and each torus divides the three-dimensional iso-energy surface into two isolated regions: a region inside the torus and another region outside the torus. Thus no trajectory can cross the torus and no excursions from one region to another occur[3]. Therefore, if the initial state of the system is enclosed by a KAM surface, the response of the system is bounded even if the iso-energy surface is not bounded.

APPENDIX B: PARAMETERS OF A THREE-MACHINE POWER SYSTEM

The parameters of an example three machine power system are listed in the following and the configuration of the system is shown in Fig.2.

The mechanical inertias of the machines are

$$\mathbf{M} = \mathrm{diag}\,[2.0, \quad 1.0, \quad 3.0] \tag{24}$$

The transmitting capacities of the transmission lines are

$$\mathbf{F} = \begin{bmatrix} 0 & 5.0 & 10.0 \\ 5.0 & 0 & 7.5 \\ 10.0 & 7.5 & 0 \end{bmatrix} \tag{25}$$

The mechanical power input to the machines are

$$\mathbf{P}^{mT} = [5.0 \;\; 2.0 \;\; -7.0] \tag{26}$$

The stable equilibrium point of the system is

$$\boldsymbol{\delta}^{sT} = [0.249 \;\; 0.137 \;\; -0.211] \tag{27}$$

The unstable equilibrium point with minimum energy is

$$\boldsymbol{\delta}^{uepT} = [2.517 \;\; 0.574 \;\; -0.107] \tag{28}$$

The energy level at this unstable equilibrium point, which is the critical energy level of the system, is $H^{uep}_{min} = 14.941$. The cutset corresponding to this unstable equilibrium point consists of the transmission lines $(1,2)$ and $(1,3)$.

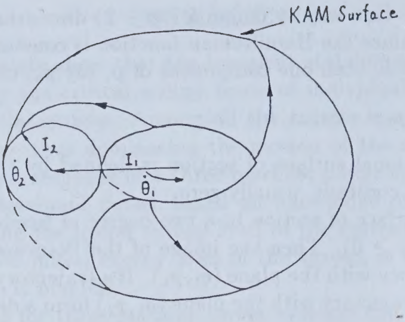

KAM Surface
I_2
I_1
$\dot{\theta}_2$
$\dot{\theta}_1$

Figure 1: A two-degree of freedom integrable Hamiltonian system can be viewed as a system consisting of two hinged rotating rods. The path of the moving end of rod 2 lies on a two-dimensional torus which is a KAM surface

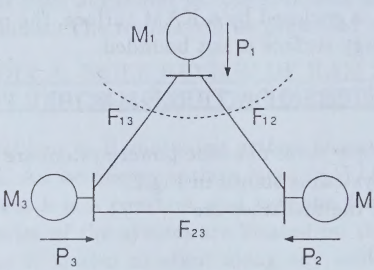

M1
P1
F13
F12
M3
M2
F23
P3
P2

Figure 2: Configuration of the three-machine power system. The cutset corresponding to the unstable equilibrium point with minimum energy is shown in dashed line.

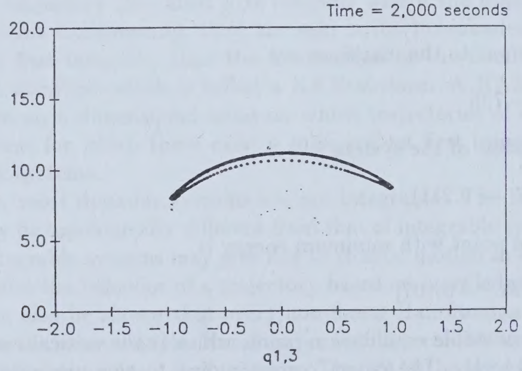


Figure 3: Image of the Poincare map with the surface of section defined by $(Q_2 = 0, P_2 \geq 0)$ $((q_{2,3} = 0, p_{2,3} \geq 0))$. Three-bus ground fault at machine 1 for 0.8 seconds; the energy level of the system is 13.398.

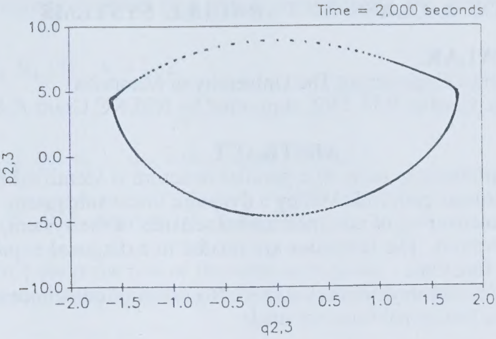


Figure 4: Image of the Poincare map with the surface of section defined by ($Q_2 = 0, P_2 \geq 0$) (($q_{1,3} = 0, p_{1,3} \geq 0$)). Three-bus ground fault at machine 2 for 1.22 seconds; the energy level of the system is 16.387.

REFERENCES

[1] V.I. Arnold. *Mathematical Methods of Classical Mechanics.* Springer-Verlag, New York, 1978.

[2] N. Kopell and R.B. Washburn Jr. Chaotic motions in the two-degree-of-freedom swing equations. *IEEE Trans. on Circuits and Systems*, CAS-29(11), Nov. 1982.

[3] A.J. Lichtenberg and M.A. Lieberman. *Regular and Stochastic Motion.* Springer-Verlag, New York, 1983.

[4] N. Narasimhamurthi, T. Guo, and N. H. McClamroch. On the necessary and sufficient condition for power system transient stability. in preparation.

[5] N. Narasimhamurthi and M.T. Musavi. A generalized energy function for transient stability analysis of power systems. *IEEE Trans on Circuits and Systems*, CAS-31(7), July 1984.

[6] M.A. Pai. *Power System Stability Analysis by the Direct Method of Lyapunov*, volume 3. North Holland Systems and Control Series, 1981.

[7] C. L. Siegel and J. K. Moser. *Lectures on Celestial Mechanics.* Springer-Verlag, Berlin Heidelberg New York, 1971.

IDENTIFICATION OF A CLASS OF PARALLEL SYSTEMS

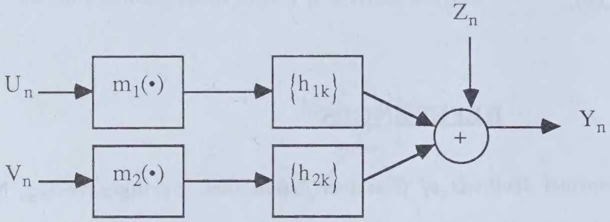
MIROSLAW PAWLAK,
Department of Electrical Engineering,The University of Manitoba
Winnipeg, Manitoba, Canada R3T 2N2, supported by NSERC Grant A-8131

ABSTRACT

In this paper a certain nonlinear system of a parallel structure is identified.Each branch of the system consists of a nonlinear gain followed by a dynamic linear subsystem.
The paper concerns the recovering of nonlinear characteristics of the system. It is done with the help of orthonormal functions. The functions are related to a diagonal expansion of the input joint probability density function.
The conditions for the consistency are established. No assumptions concerning continuity of the characteristics or their functional form are made.

I. PROBLEM STATEMENT AND PRELIMINARIES

In this paper we identify the following nonlinear parallel system.



Here m_1, m_2 are nonlinear memoryless characteristics and $\{h_{1k}\}$ $\{h_{2k}\}$ represent linear dynamical subsystems
The input U_n, V_n is stationary white noise with joint density f(u,v). The output Y_n is disturbed by a zero mean stationary white noise Z_n. The problem of the present paper is to identify separately m_1 and m_2 from the input-output data $\{(U_k, V_k) ; Y_k\}, 1 \le k \le n$.
The case when U_n and V_n are independent has been examined in [2]. For the related problem in the case of systems without memory, see [5]. Moreover, for the identification of one channel system, see [3].
Without loss of generality let $h_{01} = h_{02} = 1$ and $E\, m_1(U) = E\, m_2(V) = 0$. Next, let us assume [1], [6], that the bivariate density f(u,v) can be represented by a diagonal expansion of the form

$$f(u,v) = f_1(u)\, f_2(v) \sum_{i=0}^{\infty} c_i \theta_{1i}(u)\, \theta_{2i}(v)\,, \tag{1}$$

where f_1 and f_2 are marginal densities of f (u, v) and $\{\theta_{1i}\}, \{\theta_{2i}\}$ are complete orthonormal systems of functions in $L_2(f_1)$ and $L_2(f_2)$, respectively. Moreover, $c_i = E\{\theta_{1i}(U)\ \theta_{2i}(V)\}$.
By these conditions we can get the following system of integral equations:

$$\begin{cases} E\{Y_n | U_n = u\} = m_1(u) + \int m_2(v)\, f(v | u)\, dv \\ E\{Y_n | V_n = v\} = m_2(v) + \int m_1(u)\, f(u | v)\, du \end{cases}$$

By this and by invoking (1) we are in a position to estimate m_1 and m_2

$$\widehat{m}_k(u) = \sum_{k=0}^{N(n)} \widehat{d}_{ki}\,\theta_{ki}(u)\,, \quad k = 1, 2 \tag{2}$$

where

$$\widehat{d}_{1i} = (\hat{a}_{1i} - \hat{a}_{2i}c_i)\,/\,(1-c_i^2)\,,\ \widehat{d}_{2i} = (\hat{a}_{2i} - \hat{a}_{1i}c_i)\,/\,(1-c_i^2)\,,$$

$$\hat{a}_{1i} = n^{-1}\sum_{t=1}^{n} Y_t\,\theta_{1i}(U_t)\,,\ \hat{a}_{2i} = n^{-1}\sum_{t=1}^{n} Y_t\,\theta_{2i}(V_t)\ .$$

The sequence $\{N(n)\}$ plays the role of the truncation point.
The estimate (2) is motivated by orthogonal series estimates of regression and density functions [4].

II CONSISTENCY RESULTS

In this section we present conditions under which a global measure of the estimate performance tends to zero as $n \to \infty$.

Theorem . Let both dynamical subsystems be stable. Let $|c_i| < 1$ and $|\theta_{ki}(u)| \leq G_i$, $k = 1,2$.

If $N(n) \to \infty$ and $n^{-1}\sum_{i=1}^{N(n)} G_i^2\,(1-|c_i|)^{-2} \xrightarrow{n} 0$ then

$$E\int (\widehat{m}_k(u) - m_k(u))^2 f_k(u)du \to 0 \text{ as } n \to \infty\,,\ k = 1, 2\ .$$

Let us note that no continuity conditions for m_1 and m_2 have been made. For $f(u,v)$ being the bivariate Gaussian density with the correlation $|\rho| < 1$ the above condition resolves to

$N(n)/n \xrightarrow{n} 0$.

REFERENCES

[1] T.F. Barret and D.G. Lampard, "An expansion for some second-order probability distributions and its application to noise problem", IRE Trans. on Information Theory, Vol. IT-1, pp. 10-15, 1955.

[2] W. Greblicki and M. Pawlak, "Nonparametric identification of two-channel nonlinear systems", Proc. of the 25th IEEE Conference on Decision and Control, Athens, Vol. 3, pp. 2012-2015, 1986.

[3] W. Greblicki and M. Pawlak, " Nonparametric Identification of Hammerestein Systems", IEEE Trans. on Information Theory, 1988, to appear.

[4] W. Greblicki and M. Pawlak, "Fourier and Hermite Series Estimates of Regression Functions", Ann. Inst. Statist. Math., Vol. 37, pp. 443-459, 1985.

[5] C.T. Stone, "Additive regression and other nonparametric models", Ann. Statist., vol. 13, pp. 689-705, 1985.

[6] G.L. Wise and J. B. Thomas, "On a completeness property of series expansions of bivariate densities", Journal of Multivariate Analysis, Vol. 5, pp. 243-247, 1975.

Using Replication to Implement Reliable Storage

Bharat Bhargava and Shirley Browne
Department of Computer Sciences
Purdue University
West Lafayette, IN 47907

Abstract. A method is presented for using main memory replicated copies instead of disk-based storage to achieve reliable transaction processing. The discussion focuses on how the method can be implemented efficiently and on how it can overcome some of the drawbacks of conventional recovery techniques.

1 Introduction

1.1 The problem

We consider a distributed database system with replicated copies that supports atomic, durable transactions. Atomicity means that either all of a transaction's actions are done or none are done, and durability means that the effects of a committed transaction are guaranteed to survive [7]. The usual methods for ensuring atomicity and durability combine use of a local log on stable storage with a distributed atomic commitment process [1]. The updates of a transaction must be recorded on stable storage before the transaction is committed so that the effects of a transaction will be preserved in the event of failures (durability). Also, once the decision has been made to commit a transaction, that decision must be recorded on stable storage so that, even if site failures occur, all sites eventually reach the same decision and update their data accordingly (atomicity).

Conventional recovery techniques have the following drawbacks:

1. Commitment of a transaction involves writing one or more log records to disk. Since read and write locks are held until the transaction has committed, holding these locks during disk writes reduces concurrency.
2. There is no physical separation between the log and the data for a particular site. In the event of a disaster, both could be lost.
3. Undo/redo processing may by lengthy if the transaction rate is high compared to the checkpointing frequency.

1.2 A solution

We define *logical reliable storage* (LRS) to be storage that survives all system failures, without regard to how such storage is implemented. LRS can be implemented in the conventional way by using stable disk storage, or by recording information in physically separated replicated main memory copies. We present commit, termination, and recovery protocols that use replication to implement LRS. The commit point of a transaction is defined to be the point at which a commit record has been written to LRS. Our definition of the commit point is more general than the conventional one, in which the commit point occurs as soon as the coordinator of the transaction has written a commit record to disk.

For simplicity, we at first assume that data are fully replicated, that only site failures occur, and that strict two-phase locking is used for concurrency control. Our scheme can be extended to handle partial replication the use of other methods for concurrency control. Strategies for dealing with network partitioning failures are discussed in section 3.4.

Our method is intended for use either with a main-memory database (in which the entire database can fit into virtual memory) [5] or with a conventional database that uses caching. In any case, we assume that read and write operations access only main memory whenever possible and that disk writes are asynchronous to transaction processing.

We assume that each site has probability p of failing and that site failures are independent of each other. To ensure that the probability of an update being lost is less than q, the update must have been recorded at greater than or equal to a threshold number of sites T, where $p^T \leq q$. The commit point of a transaction occurs as soon as T sites have received the commit message Thus, for large enough T, the likelihood of an update or a commit record being lost once it has been recorded at T sites can be made as small as the likelihood of log failure in the case of a conventional logging scheme.

In section 3.1, we present a simple version of our recovery scheme in which a site recovering from a failure must copy the entire database from an operational site. Although this recovery technique might be appropriate for a small database, it will be too inefficient for a large database system. We then describe, in sections 3.2 and 3.3, modifications that allow for more efficient recovery. Extensions to deal with network partitioning are discussed in section 3.4.

2 Commit and Termination Protocols

2.1 Commit protocol

Atomicity is achieved by performing transaction commitment in two phases. In our model, the first phase consists of recording the updates at a threshold number T of the sites. In the second phase, the updates are applied to the database copies, i.e., the results of the transaction are made visible. Since we assume that not all T sites will fail, the second phase can be restarted at any site by communicating with one of these T sites. A more detailed description of the two phases is given below.

When the coordinator of a transaction finishes the execution of the transaction, it makes a decision as to whether to commit or abort the transaction. If slave processes have the right to unilaterally abort at this point (true in the case of deferred writes, for example), then the coordinator sends a round of vote request messages and waits for replies before making the decision. After reaching a decision, the coordinator broadcasts that decision to all sites (including itself). At this point, the commit protocol becomes decentralized.

If a site receives an *abort* message, it undoes any updates made by the transaction (by using before images) and releases all the transaction's locks. If a site receives a *commit* message, it sends *commit* messages to some designated subset of the other sites and waits until it has received *commit* messages from at least $T-1$ sites. At this point the site can commit the transaction. It applies any updates that have not yet been done and releases all the transaction's locks. If a site fails to receive $T-1$ *commit* messages, it may query other sites to make up the deficiency. If it still fails to receive $T-1$ *commit* messages or if it has not received any *commit* messages and detects that the coordinator site has failed, it can invoke the termination protocol.

2.2 Termination protocol

To invoke the termination protocol, a site broadcasts either a *noncommittable* message (if it has not received any *commit* messages) or a *committable* message (if it has received at least one *commit* message). Subsequent rounds of the termination protocol are the same as in [17], with the following exceptions: A site commits a transaction after receiving a round that has at least T *committable* messages. Once a site has committed a transaction, it sends a *committed* message in all subsequent rounds. On receiving a single *committed* message, a site commits the transaction. If two rounds of *noncommittable* messages are received from the same set of sites but T or more sites have failed (a situation that is not supposed to occur!), the transaction is blocked (since the T failed sites may have committed the transaction).

A question arises as to what action should be taken by a site if it knows that several sites have already failed. For example, suppose a site receives a single *committed* message and all other messages in the same round are *noncommittable*. The site knows that $T-1$ sites that received *commit* messages have failed (Suppose T is large, say greater than ten). It also knows that two more site failures will be disastrous (Again, this situation is not supposed to occur! The situation is similar to having tossed a fair coin eight times and gotten eight straight tails. Still the probability of getting a tail on the next toss is one half). One strategy would be to force a back-up log for that transaction to disk so that a record of the transaction is saved. Then if all sites that have committed the transaction fail before informing the remaining sites, the transaction will be blocked until some site that wrote the transaction record to disk recovers. Nevertheless, the updates will not be lost unless no site was able to write the transaction record to disk.

3 Recovery

A site may start again to participate in transaction processing as soon as it is up and running. Lock and write requests may be honored immediately. Read requests cannot be honored, however, until the local copy has been refreshed or it has been determined that the local copy is up-to-date. The ability to determine whether or not a copy is up-to-date depends on the version of recovery used, as described below.

3.1 Simple version using copying

The simple version of our recovery method requires a recovering site to copy the entire database from an operational site. The recovering site accomplishes this by running a copier transaction for each data item. The copier transaction reads an up-to-date copy at an operational site and uses the value read to renovate the local copy. As suggested in [4], a two-stage recovery period may be used: during the first stage, copies are refreshed by normal transaction processing and by the recovering site requesting copier transactions on demand; during the second stage, the recovering site issues copier transactions in a batch mode. To ensure logical-serializability [15] of user transactions, we require the coordinator of a user transaction to check that any sites that were failed when write locks were requested are still failed once the lock point of the transaction has been reached. With this condition imposed, it is easy to show that user transactions are logical-serializable in the order of their lock points. The proof consists of showing that the execution history for the replicated database has the same read-from relations as a serial one-copy history with transactions in lock-point order.

3.2 Recovery using in-memory logs

For reasons of efficiency, we may want to avoid having a site copy the entire database upon recovery. Our strategy for efficient recovery borrows from the conventional approach of using a log and taking checkpoints, with the differences that logs are kept in memory and checkpoints are approximate.

Each site keeps a sequential log in memory to which it appends a record of every operation that is performed on behalf of a transaction. The operations include begin, read, write, get lock, release lock, vote, commit, and abort. The log is physically locked by a transaction only for the duration of a single operation. Thus, the operation and the update to the log are performed atomically, and the union of all the in-memory logs composes a logical-serializable history [15]. The log at a particular site will contain checkpoints for each of the other sites. A method for setting these checkpoints is described later.

To recover from failure, a site carries out the following steps:

1. locks the log at an operational site, appends a marker, and unlocks the log.
2. scans the log from its checkpoint up to its marker. All items found to have been written by a committed transaction or locked by an active transaction are marked as unreadable at the recovering site. An exception to this rule occurs if a data

item has already been refreshed by a transaction update. This technique is similar to the use of fail locks, mentioned in [2].

Again, a formal proof can be given to show that the resulting execution history is equivalent to a one-copy serial history with transactions in lock point order. Intuitively, we can reason that a site will not miss any updates as follows: If a transaction coordinated by some other site has not yet reached its lock point at the time that the recovering site accesses the remote log, this transaction will either abort or request locks from the recovering site. If the transaction has already reached its lock point, any data items updated by the transaction will be either marked unreadable or updated at the recovering site.

For any active transactions for which it was the coordinator, the recovering site effectively "forgets" that it was the coordinator. Thus, if the active transaction had not yet reached the point where the commit protocol becomes decentralized, the transaction will be aborted. If the decentralization point had been reached, then the transaction will be committed or aborted according to the termination protocol.

It may be undesirable to restart a long-running transaction from the beginning following coordinator failure, since a large amount of work may have to be redone. An alternative, discussed in [8] and [13], is to provide transaction save points or checkpoints, so that a transaction need only be rolled back to some previous saved state in the event of a failure. Our scheme can be extended to incorporate such ideas.

3.3 Checkpointing in main memory

To be able to discard obsolete log information and to avoid having a site process an entire log upon recovery, we use a checkpointing method. Similar methods are developed in [16] and in [9] for the purpose of discarding obsolete history information. Each site maintains a set called DONE in which it stores the TID's (transaction ID's) of all recently committed and aborted transactions. (We assume a new TID is assigned at a coordinating site by incrementing a counter and appending a unique site ID to the new counter value. Hence, TID's are systemwide unique and increasing over time at any particular site. Although not strictly required for correctness, for reasons of space efficiency counters should ideally be logical clocks [11]). Each site also maintains a set called ACTIVE in which it stores information concerning the active transactions at that site. (A transaction is active if it has not yet been committed or aborted).

Each "dirty" data item has a pointer to the TID in DONE corresponding to the transaction that most recently modified that data item. If the TID corresponds to a committed transaction, then the update for that transaction has been applied to the data item. If the TID corresponds to an aborted transaction, then the update has been undone. Hence, once the dirty data item has been written to disk, the log entry for that transaction will no longer be needed. With each TID in DONE is associated a reference count that gives the number of dirty data items that point to the TID. Whenever a dirty data item is written to disk, the reference counts of all the TID's it points to are decremented. When the reference count for a particular TID becomes zero, that TID may be deleted from DONE.

Each site maintains a checkpoint value that is the minimum of the following:

1) the counter values most recently received from the other sites (the value sent out by a site for its counter must be less than the TID assigned to any active transaction by the site),

2) the site's own counter value,

3) TID's of all transactions in ACTIVE, and

4) TID's of all transactions in DONE.

A site will never need the log record for any transaction that has a TID smaller than its checkpoint value. Whenever a site is able to increase its checkpoint value, it may broadcast the new value, possibly by piggybacking it on top of some other message. Upon receiving a checkpoint value from another site, say site A, a site may move the checkpoint for site A forward in the log until it encounters any entry with a TID greater than or equal to the checkpoint value. Log records may be deleted up to the smallest checkpoint. This processing of the log may be done simultaneously with another log access.

3.4 Recovery from Network Partitioning

We assume that a quorum-based method is used to ensure consistency in spite of the possibility of network partitioning. If a static quorum method is used, our simple recovery method can be easily extended. A recovering site will either know the static quorum assignments (i.e., they are recorded on permanent disk storage and restored during startup) or can obtain them from an operational site. Any write quorum must be at least as large as T, our threshold number of sites. A site cannot be part of a read quorum for a data item if that item is marked as unreadable, but it can be part of a write quorum for such an item. A copier transaction must access a read quorum for the data item it is renovating.

If dynamic quorum adjustment [10] is used, greater care must be taken to ensure that a recovering site knows which quorum assignments are currently in effect. Also, several versions of a data item may be in use at one time, and the recovering site may refresh all or only some of these versions. In the method presented in [10], the transaction coordinator chooses a level in which to execute a transaction. Each data item has a quorum assignment table that assigns a read quorum and a write quorum for each level. Dynamic quorum adjustment is carried out by means of transactions that treat quorum assignment tables as data. A recovering site must obtain a copy of a valid quorum assignment table before it can begin to coordinate transaction processing or refresh any data items. One solution is to require that all updates to quorum assignment tables be logged to stable disk storage. Then a recovering site will obtain valid quorum assignment tables during the startup process. Since updates to quorum assignment tables should occur infrequently, the overhead of logging such changes should not be too great. An alternative solution is to have a recovering site copy the quorum assignment tables from an operational site. The recovering site cannot directly use these tables, however, since they may have been changed by writing them to coquorums for read and write that included the recovering site itself. Hence, for a given level, the recovering site must access a coquorum for the read and write quorums given in the copied table to be assured of obtaining a valid quorum assignment for that level. Since requiring access to these coquorums may severely restrict availability, it seems better to log quorum assignment updates to stable disk storage.

3.5 Comments

We have omitted discussion of several complexities that arise from the use of in-memory logs and in-memory checkpointing. For example, how should the replicated log itself be refreshed at the recovering site? We could either copy the log from an operational site or start anew. If we start a new log from the point of recovery, then we will need to determine a time after which this log can be accessed by a subsequent recovering site. There are also some subtleties concerning when a recovering site can begin to send checkpoint values. It must rebuild its DONE data structure. Also, it cannot send a checkpoint value as long as any items are marked unreadable.

In-memory logs do not appear to be of much use in the context of network partitioning, since it is difficult to say how such logs should be merged upon recovery from partitioning. If we assume, however, that partitioning is relatively infrequent compared to site failure and that partitions are long-lived, then in-memory logs could be used within a stable partition in a manner analogous to the case where we have only site failures.

4 Conclusions

We have presented commit and recovery protocols that use replication at distributed sites in place of disk-based logging to achieve reliability. We have defined the commit point of a transaction to be the point at which a commit message has been received by a threshold number T of the sites. This definition differs from the conventional one, in which a transaction reaches its commit point immediately after a commit record has been written to disk by the transaction coordinator. We base our method on the assumption that the T sites that have received the commit message will not all fail. Thus, our method is not applicable to a local area network in which all sites may fail at once.

Our solution addresses the points raised in the introduction as follows:

1. During normal transaction processing, locks on data items are not held during disk writes, since writes to disk are asynchronous to transaction processing.
2. Data are replicated at all sites. Thus, even if the contents of both volatile memory and disk storage at a particular site are lost, recovery is still possible.
3. Undo/redo processing of an old value/new value log can be avoided by copying the entire database or by using a log to determine the set of modified items and copying only these items.

Since commit and checkpoint processing are typically I/O bound, our techniques should give better performance than conventional methods in some situations. The conventional commit protocol requires communication anyway; hence, under full replication, our commit protocol has essentially no additional communication overhead. With partial replication, however, communication overhead for our method may be greater. In the two-phase commit protocol implemented in Argus [12], the minimal delay of user code (at the coordinating site) for committing a top-level transaction is one network round-trip (requiring 35 ms) plus two writes to stable storage (at 17 ms each, although the author notes that implementation of true stable storage would require two sequential writes per disk write, or double the cost). Minimal delay of user code for our scheme would be two network round-trips, about the same time as for the Argus implementation. We should note, however, that the estimation of minimal delay for Argus assumes that all updates at participant sites have already been written to stable storage in the background before the prepare message (sent by the coordinator in the first phase) arrives. A comparison using the performance numbers given in [5] shows our scheme to be superior to a conventional logging scheme: round-trip network communication costs are given as 8 ms, while the time for the disk write of the log entry is given as 20 ms. For the V distributed system [6], round trip network communication times for sending a 32-byte request and receiving a 32-byte response are cited as 2.54 ms for 0 Kbytes of data and 17.8 ms for 8 Kbytes. For our commit protocol, assuming the first phase messages fit into 8 Kbytes, the communication costs in V would add up to 20.3 ms. In [6] it is also stated that the performance of interprocess communication in V is such that it is faster to access a copy of a disk block at a remote server that has the data in its RAM cache than to read it from a local disk. Such a situation would indicate that our remote in-memory logging scheme would provide faster recovery than conventional logging, assuming the remote log had not been paged out to disk. The figures cited above are for local area networks. In the case of long-haul networks, round trip communication times are much greater. However, replication allows the continued processing of transactions when a site is down; thus, undo/redo processing of a local log will be of little or no value if the site has been down for a long time, since the local disk-based log will not have any record of updates that were missed while the site was down.

Recent research has focused both on improving communications performance [3] and on improving I/O performance [14]. The impact of these advances on the relative efficiency of various commit and recovery protocols merits investigation.

It is shown in [18] that a centralized nonblocking commit protocol requires at least three phases under the assumption that any number of sites may fail. Our commit protocol should allow for faster nonblocking commitment, since a site does not have to wait for all sites to reach a prepared state, as in [18], before it can commit the transaction. Of course, for our protocol to be nonblocking, we require that any T sites that have received the commit message cannot all fail.

For future work, we intend to extend our methods to handle network partitioning failures. We also intend to implement our scheme and compare its performance with that of conventional recovery methods under both full and partial replication.

References

[1] P. Bernstein, V. Hadzilacos, and N. Goodman. *Concurrency Control and Recovery in Database Systems.* Addison-Wesley, 1987.

[2] B. Bhargava. Transaction processing and consistency control of replicated copies during failures in distributed databases. *Journal of Management and Information Systems*, 4(2):93–112, Fall 1987.

[3] B. Bhargava, T. Mueller, and J. Riedl. Experimental analysis of layered ethernet software. In *Proc. ACM-IEEE Computer Society Fall Joint Computer Conference*, pages 559–568, Dallas, Texas, Oct. 1987.

[4] B. Bhargava, P. Noll, and D. Sabo. An experimental analysis of replicated copy control during site failure and recovery. In *Proc. 4th IEEE Data Engineering Conference*, pages 82–91, Los Angeles, Feb. 1988.

[5] A. D. Birrell, M. B. Jones, and E. P. Wobber. *A simple and efficient implementation for small databases.* Technical Report 24, Digital Systems Research Center, Palo Alto, CA, Jan. 1988.

[6] D. R. Cheriton. The V distributed system. *Commun. ACM*, 31(3):314–333, March 1988.

[7] J. Gray. The transaction concept: virtues and limitations. In *Proc. 7th Int. Conf. Very Large Databases*, pages 144–154, IEEE, 1981.

[8] J. N. Gray, P. McJones, M. Blasgen, B. Lindsay, R. Lorie, T. Price, F. Putzolu, and I. Traiger. The recovery manager of the system R database manager. *Computing Surveys*, 13(2):223–242, June 1981.

[9] A. Heddaya, M. Hsu, and W. Weihl. *Two phase gossip: managing distributed event histories.* Technical Report TR-04-87, Aiken Computation Lab., Harvard University, 1987. To appear in Information Sciences: An International Journal in 1988.

[10] M. Herlihy. Dynamic quorum adjustment for partitioned data. *ACM Trans. Database Syst.*, 12(2):170–194, June 1987.

[11] L. Lamport. Time, clocks, and the ordering of events in a distributed system. *Commun. ACM*, 21(7):558–565, July 1978.

[12] B. Liskov. Distributed programming in Argus. *Commun. ACM*, 31(3):300–312, March 1988.

[13] J. E. B. Moss. Checkpoint and restart in distributed transaction systems. In *Third Symposium on Reliability in Distributed Software and Database Systems*, pages 85–89, IEEE Computer Society Press, Oct. 1983.

[14] D. A. Patterson, G. Gibson, and R. H. Katz. A case for redundant arrays of inexpensive disks (RAID). In *Proc. SIGMOD '88 International Conference on Management of Data*, pages 109–116, June 1988.

[15] Z. Ruan. *File replication in distributed systems.* PhD thesis, Purdue University, Aug. 1986.

[16] S. Sarin and N. Lynch. Discarding obsolete information in a replicated database system. *IEEE Trans. on Software Engineering*, SE-13(1):39–47, Jan. 1987.

[17] D. Skeen. A decentralized termination protocol. In *Proc. 1st IEEE Symp. on Reliability in Distributed Software and Database Systems*, pages 27–32, July 1981.

[18] D. Skeen. Nonblocking commit protocols. In *Proceedings of the 1981 ACM-SIGMOD International Conference on Management of Data*, pages 133–142, Ann Arbor, Michigan, 1981.

CORRESPONDENT COMPUTING BASED SOFTWARE FAULT TOLERANCE

PEN-NAN LEE, ADERBAD TAMBOLI & JEFF BLANKENSHIP
Department of Computer Science, University of Houston
Houston, Texas.

ABSTRACT

In this paper, we propose a new methodology based on Correspondent Computing to achieve error detection, and to assist in error recovery. Correspondent Computing is the generation of effects which are correspondent to the effects of an operation. On the basis of these correspondent effects, a Comparative Test is formulated, using the precise relationship between an operation and its correspondent operations to perform error detection. In addition, these correspondent operations provide the necessary redundancy for forward error recovery. These features make this concept an effective and viable tool in software fault tolerance.

1. INTRODUCTION

The extensive use of computer systems in high reliability and availability applications has manifested the need for Software Fault Tolerance. Since, in software development, residual design and implementation inadequacies are possible, the execution of a software may not always result in the desired output. To ensure the desired results, software fault tolerance techniques can be used to provide the ability to detect and recover from these errors. Software fault tolerance is one of two complementary approaches for enhancing software reliability. The first, software fault prevention is used to promote error prevention, detection and removal prior to placing the software on line. Fault tolerance, on the other hand, requires error detection and recovery capabilities during execution to prevent faults from causing failures due to residual design inadequacies.

In software fault tolerance, a simple replication of components cannot be used to provide redundancy. Rather, these software components must be developed and implemented using "different design". Otherwise, these redundant components would suffer from the same errors and fail under the same circumstances. These software modules are created from the same initial design specifications, using various techniques, including different programming languages, algorithms, programmers etc., to achieve the desired redundancy. There are two strategies for providing a fault tolerant environment. These are the Recovery Block [RAND75], which involves the sequential execution of alternate modules upon failure of the primary module. The second, called N-Version Programming [CHEN78], consists of parallel execution of all modules in a concurrent environment.

The Recovery Block strategy uses an acceptance test, which is a logical expression evaluated on exiting any alternate, to determine whether the alternate has performed acceptably. A further alternate, if one exists, is entered if the preceding alternate fails the acceptance test, or fails to complete. A disadvantage of the acceptance test is that it provides a means to check "reasonableness" and not "correctness". The more precise the acceptance test, the more difficult it is to provide, as there is no automatic method to provide specifications. In N-Version programming, redundancy demands that the N (>=2) different version be totally independent, yet must obey the same initial design specifications. All versions start simultaneously, and execute concurrently. Upon termination of all the versions, a "desired" result is then "elected" using "majority voting". A drawback of N-version programming is the use of inexact voting, which requires some form of "discrepancy range" to be provided. This may be difficult to provide and requires knowledge of the data, underlying machine and the tolerable range. If not correctly formulated and implemented these error detection components themselves might be intolerant to faults.

In this paper, we propose a new methodology based on Correspondent Computing [LEE88] to achieve error detection, and to assist in error recovery. Correspondent computing is the generation of effects (or results, outcomes) which are correspondent to the effect of an operation. Correspondent computing provides an elegant means of useful redundancy. This redundancy is not only simple to formulate, but also powerful enough to assist in error detection and error recovery. Error detection is provided through the use of a Comparative test, which is based on the simple and precise relationship of the primary operation and its correspondent operations. These operations provide useful redundancy and therefore have the prerequisites for applications in fault tolerance. In addition to providing a methodology for software fault tolerance, correspondent computing also provides a complimentary approach for the selection of alternates and versions used in the Recovery Block and the N-version programming strategies.

The concept of correspondent computing is further refined and developed in section 2. In section 3, we discuss error detection and recovery strategies using correspondent computing. Section 4 provides examples of implementing software fault tolerance using correspondent computing. Section 5 compares correspondent computing with recovery blocks, and N-version programming. Finally, in section 6, we discuss the impact of correspondent computing, and its future implications in software fault tolerance.

2. CORRESPONDENT COMPUTING

The theory of Correspondent Computing is based on the philosophy of correspondence. Which can be stated as "if executing an operation can produce a significant effect, an equally significant effect can be generated by another semantically correspondent operation". The object of Correspondent Computing is to generate results which exhibit correspondent relationships.

The use of correspondence relationships for error detection appears in a number of application areas. For example, in business the double entry bookkeeping system is a method to ensure that credits and debits ultimately balance. Suppose Company A takes a loan of $5,000 from Bank B. Therefore an asset item (cash $5,000) appears in the credit column of Company A. There is a corresponding liability item (loan $5,000) in the debit column as Company A owes Bank B. Both assets and liabilities increase correspondingly, hence the balance is maintained. On the other hand, Bank B has an asset item (loaned $5,000) in its credit column and a corresponding asset item in the debit column (less cash $5,000). Therefore, the assets have increased by $5,000 (loan creditor) as well as decreased correspondingly by $5,000 cash. Hence, equality of assets is maintained throughout.

An operation may be defined as a single computer instruction, or a sequence of such instructions. In effect, we will use the term operation to denote any such condition. Correspondence is the property that binds two operations together. The relationship between the two operations being equivalent, reciprocal or analogous. Two operations are semantically correspondent if the same correspondence relationship exists for the operations and their effects. That is, the relationship between the effects of the two operations is exactly the same as the relationship between the two operations.

For example, two operations A and B produce effects A' and B' respectively. Let the relationship between the effects A' and B' be inverse i.e. A' is the inverse of B' ($A' = 1/B'$). Then the two operations A and B are semantically correspondent if A is the inverse of B ($A = 1/B$).

operation	A : $x = d / c$	B : $y = c / d$
effect	A' : $x = 4$	B' : $y = 1 / 4$

Considering the example above the relationship between A' and B' is inverse : $A' = 1/ B' = 1/ (1/4) = 4$ and the relationship between A and B is also inverse : $A = 1 / B = 1 / (c / d) = d / c$. Consequently operations A and B are semantically correspondent operations.

Consider another example, a force F is acting on an object, and the object produces a reaction R. Now consider that we have another force F' acting on the same object and producing reaction R'. If we consider the forces as operations then the reactions are the effects. We can say that F and F' are correspondent operations, since they both provide a similar effect (i.e reaction) and the magnitude of these effects differ in the exact proportion that the magnitudes of F and F' differ from each other.

The operation whose effect is the target of correspondent computing is called the primary operation. The primary operation can be thought of as an operation whose correct execution is more critical than other constituent program operations. Operations which generate effects that correspond to the primary operation are called correspondent operations. The effects of a correspondent operation may be equivalent, complementary, contradictory, or competing to the effects of the primary operation. Correspondent operations can be categorized as Reciprocal, Duplicate, or Residual. A reciprocal operation is defined as a correspondent operation whose behavior is semantically and syntactically the inverse of the primary operation. A duplicate operation is a correspondent operation which is syntactically and semantically equivalent to the primary operation, or in other words it is simply a copy of the primary operation. An operation that fulfills the definition of correspondent operations, but is neither a duplicate, nor a reciprocal is termed as a residual operation. A residual operation is difficult to conceive at first, but is the most powerful and versatile of the three. Let us formulate correspondent operations for a simple primary operation.

Primary operation	P :	p := b / c ;
Reciprocal operation	R :	r := c / b ;
Duplicate operation	D :	d := b / c ;
Residual operation	RE :	re := b * c ; or re := b + c ;

Consider an actual program example.

Primary operation	P	: Sort an array of integers in ascending order.
Reciprocal operation	R	: Sort the array in descending order.
Duplicate operation	D	: Sort in ascending order.
Residual operation	RE	: Sort the indices of the array in ascending order instead of the elements.

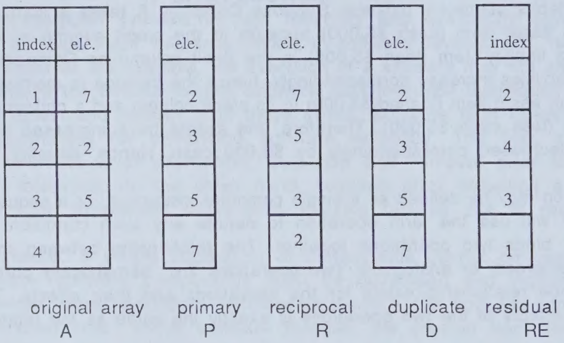


Fig. 1 : Snapshot of the arrays before and after correspondent operations.

Fig. 1 shows a snapshot of the arrays. Notice that in the residual operation we sort the array and store the index of the sorted array, thus to access elements in the residual array in ascending order we do : A [RE [i]] : i := 1.. n ;. Computation errors and data inconsistency errors can be checked by comparing the primary array with the correspondent arrays.

For any primary operation, its duplicate operation always exists and is the simplest to formulate. But it suffers from the same design inadequacies as the primary. On the other hand reciprocal operations lend themselves very naturally to arithmetic primary operations, but under certain circumstances are very difficult to specify. Whereas, for every primary operation it is possible to formulate a residual operation.

At this point we introduce the concept of "designated algorithm". The algorithm that is implemented, is termed as the designated algorithm, and is chosen from among all algorithms that satisfy the specifications. The basic requirements for selecting a designated algorithm includes : 1) careful inspection of the algorithm, to ensure that the algorithm satisfies the specifications completely, and 2) a designated algorithm must be provably correct, that is, using some formal verification method, the algorithm must be proved to be mathematically correct. If all the selected algorithms meet the above basic requirements, some optional selection criteria can be used to select the designated algorithm. They are 1) time and space efficiency, 2) the algorithm should be implementable on the system to be used, and 3) the algorithm must be decomposable into its constituent parts. Decomposable implies that the algorithm can be broken down into "operations".

The primary operation is critical to the correct functioning of the software, and is the basis for the implementation of correspondent operations. Obviously, this requires that the designated algorithm be used as the primary algorithm. Once selected, this primary (designated) algorithm can also serve as the target algorithm for generating "correspondent" algorithms. The correspondent algorithms derived are from the same designated algorithm, and are not generated independently using totally different algorithms. This makes it easier to provide the necessary redundancy. Since the primary algorithm has already been proved correct, only residual errors in the implementation of the primary can occur. However, design and implementation errors can occur in the correspondent versions, if the correspondent operations are not carefully formulated. Any other form of error in the primary, specification mismatch errors, indicate that the correct designated algorithm was not selected. This limitation implies that the correspondent algorithms are also incorrect, and under such circumstances, it is difficult to provide a completely fault tolerant environment. However, if the correct designated algorithm is selected, the correspondent algorithms are much simpler to formulate, and to a certain degree more error free, than independently derived at versions/alternates.

3. ERROR DETECTION AND RECOVERY STRATEGIES

The key to any fault tolerant system is its error detection mechanism, which in Recovery blocks is provided by acceptance tests. While in N-version programming voting checks perform error detection. Correspondent computing provides a simpler error detection scheme, based on Comparative testing.

Comparative testing consists of comparing the results (effects) generated by the primary operation and the results of the correspondent operation. The specification of the comparative test is based on the precise relationship between the primary and correspondent operations. We know on the basis of correspondence, that the precise relationship of the two effects should match that of the two operations. Comparative testing simply checks the relationship of the two effects. If an error occurs, the relationship between the two effects would not match that of the two operations, and the comparative test would fail.

The greatest advantage of this error detection scheme is the simple and concise specification for the comparative test. Since these specifications can be derived easily, there is a possibility that the specifications might even be generated automatically based on the correspondence relationship of the primary and its correspondent operations. Another advantage of the comparative test is that it can be performed at any time during execution of the program. Which is unlike the acceptance test and the voting check, both of which can be performed only upon exiting from the alternate / version.

From the viewpoint of software fault tolerance, the correct behavior of some operations is more critical than others, although every operation is a constituent part of the program. It is reasonable to perform error detection only on these critical operations, otherwise we would be burdened with a tremendous overhead.

The primary objective of any fault tolerant environment is the ability to recover from faults without any appreciable loss in computation or availability. Error recovery in correspondent computing can be performed by three schemes. The first scheme is purely sequential , the second and the third are concurrent variations of the first scheme.

The sequential scheme (SS) provides error detection and recovery capabilities. Redundancy is provided in the form of correspondent operations for every primary operation. The primary and zero or more correspondent operations are executed sequentially. The sequence of the operations is shown in Fig. 2.

```
P1 ; P1C1 ; .......... ; P1Cn ;
P2 ; P2C1 ; .......... ; P2Cn ;
              .
              .
Pm ; PmC1 ; ......... ; PmCn ;
```

Fig. 2 : Structure of Sequential Scheme for Correspondent Computing.

The subscript for the Pi is the i th primary operation. The subscript for Cj is the j th correspondent mechanism. That is P1C2 implies the second correspondent mechanism (e.g. reciprocal) for the first primary operation, and P2C2 implies the second correspondent mechanism (e.g reciprocal) for the second primary operation. Essentially what we are saying is that we execute a primary and then its correspondent operations before we execute the next primary operation. Error detection is provided by the comparative test, which can be performed after every sequence of primary and correspondent operations, or after a number of such sequences, or at the end of the whole program. The number of correspondent operations is variable and depends upon the criticalness of the primary operation. It is obvious that a more critical operation should have a higher degree of redundancy than a trivial operation. Thus we suggest that the number of correspondent operations for every primary depend on the importance of the primary operation. There is no need for every primary operation to have the same number of correspondent operations. Similarly, some primary operations might not require any correspondent operation.

The sequential scheme above may be inefficient in terms of execution time, as for every primary operation we have zero or more correspondent operations. To increase the efficiency of this scheme we suggest a variation, termed as Concurrent Scheme 1 (CS1). This scheme will be the base model for all other concurrent schemes. In this scheme, a primary and its correspondent operations are executed concurrently. The sequence of operations is shown in Fig. 3.

$$
\begin{array}{c}
[\, P_1 \;||\; P_1C_1 \;||\; P_1C_2 \;||\; \ldots\ldots \,]\,; \\
[\, P_2 \;||\; P_2C_1 \;||\; P_2C_2 \;||\; \ldots\ldots \,]\,; \\
\vdots \\
[\, P_m \;||\; P_mC_1 \;||\; P_mC_2 \;||\; \ldots\ldots \,]\,;
\end{array}
$$

Fig. 3: Structure of Concurrent Scheme 1 for Correspondent Computing.

Here we see that every sequence of operations has to wait until the preceding sequence finishes. This provides an appropriate position to check for errors if desired. Error detection is again provided by the comparative test, and as suggested can be used either at the end of the program, or at various points inside the program. For example, after every sequence of primary and correspondent operations, or after every x number of such sequences.

In highly concurrent applications, a variation of the base model CS1 scheme can be implemented. This technique, concurrent scheme 2 (CS2), employs two or more versions. Each version contains the primary, its correspondent operation and a comparative test, or, two correspondent operations and a comparative test. Fig. 4 shows a graphic representation of the structure of the scheme. A version upon termination will return two values, the result and a boolean value Status equal to the result of the comparative test. All the versions start simultaneously and are executed concurrently. In each version, the primary and the correspondent operations (or the two correspondent operations) are executed in parallel. The comparative test can be performed at any time during execution of the version. If the comparative test fails (at any point), then the status value is set to false and passed forward. Execution of the version is aborted when the test fails. The first version that returns a value of Status = true (i.e the comparative test has passed), is accepted as the correct version, and its result is used. On receiving a correct result all the other versions terminate instantly, or are ignored and their results are not considered. This scheme is more efficient than pure N-version programming, as all slower executing versions are ignored.

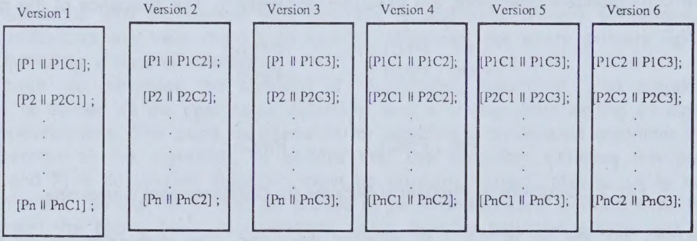


Fig. 4 : structure of Concurrent Scheme 2. Versions 1-6 are executed concurrently

Note that in Fig. 4, if any of the version consisting of the primary operation and its correspondent operations (e.g. versions 1-3) return status true , then the result of the primary will be used. Otherwise, if any one of the remaining versions (e.g. versions 4 - 6) returns status true, then this result is to be used as the correct one. However, we must convert this correspondent result into the primary result before it can be used.

4. IMPLEMENTATION OF CORRESPONDENT COMPUTING

In this section we illustrate the implementation of Correspondent computing using the simple example of Program SORT. This program sorts an array of N integers in an ascending order. A contains the original array. There are four redundant arrays, one for the primary, and one each for the three correspondent operations. All sorting is done on the individual arrays, that is , primary operation is performed on array P, duplicate operations on array D, reciprocal operations on array R and residual operations on array RE. The residual array RE contains the index of the original array , such that accessing RE in order will give the ascending sorted array. e.g to access the elements A[RE [i]] : i = 1..n ;. Fig.1 gives a snapshot of the arrays after execution completes.

4.1 SEQUENTIAL SCHEME

The program SORT is implemented using a sequential decomposition as shown in Fig. 2.

```
Program SORT ;
begin
  For i = 1 to n-1 do
    Pi   => Primary operation : Swap the ith element with the smallest element in the range i to n ;
    PiC1 => Duplicate Corr. op. : Swap the ith element with the smallest element in the range i to n ;
    PiC2 => Reciprocal Corr. op. : Swap the ith element with the largest element in the range i to n ;
    PiC3 => Residual Correspondent op. : Swap the index of the ith element with the index of the
                index of the smallest element in the range i to n ;
    Comparative test (* optional *) ;
   End ;
      Comparative test ; (*   Computation check and Data consistency check *)
    Compare P and D
      If error then compare P and R
        If error then compare P and RE
         If error then compare D and R
           If error then compare D and RE
             If error then compare R and RE
                 If error then unrecoverable error ;
End. program SORT
```

4.2 CONCURRENT SCHEME 1

In this section we implement the SORT program using the concurrent decomposition as shown in Fig. 3.

```
Program SORT ;
begin
   For i = 1 to n do
            Cobegin
                 Pi || PiC1  || PiC2 || PiC3   ;
            Coend ;
         Comparative test (* optional *)
   end ;
   Comparative test ;
End. Program SORT.
```

The comparative test and Pi, PiC1, PiC2, PiC3 are the same as in example 4.1

4.3 CONCURRENT SCHEME 2

Implementing the program SORT using concurrent scheme 2. Decomposition is as shown in Fig. 4.

```
VERSION 1
begin
   status = true ;
   For i = 1 to n do
```

```
        Cobegin
                        Pi || PiC1 ;
         Coend ;
              Comparative test  possible at this point ;
      End ;
      Comparative  test  ;
         Compare array P and array D
              If error  then status = false ;        (*  error detected *)
      return  status , P , D ;
   End ;  Version 1

   VERSION 2
   begin
      status = true   ;
      For i = 1 to n do
        Cobegin
                        Pi || PiC2  ;
        Coend ;
      End ;
      Comparative  test  ;
          Compare array P and array R
              If error  then status = false ;        (*  error detected *)
      return  status , P, R ;
   End ;  Version 2

   VERSION 3
   begin
      status = true   ;
      For i = 1 to n do
        Cobegin
                        Pi || PiC3  ;
        Coend ;
      End ;
      Comparative test  ;
          Compare array P  and array RE
               If error  then status = false ;       (*  error detected *)
      return  status , P, RE ;
   End ;  Version 3


   VERSION 4
   begin
      status = true   ;
      For i = 1 to n do
        Cobegin
                        PiC1 || PiC2  ;
        Coend ;
      End ;
      Comparative  test  ;
         Compare array D  and array R
              If error  then status = false ;        (*   error detected *)
      return  status , P, RE ;
   End ;  Version 4


   VERSION 5
   begin
      status = true   ;
   For i = 1 to n do
     Cobegin
               Pi C1 | |  PiC3  ;
     Coend ;
   End ;
   Comparative  test  ;
    Compare array D  and array RE
        If error  then status = false ;      (*   error detected *)
   return  status , P, RE ;
End ; Version 5

VERSION 6
begin
   status = true   ;
   For i = 1 to n do
     Cobegin
               PiC2 | |  PiC3  ;
     Coend ;
   End ;
   Comparative  test  ;
      Compare array R  and array RE
           If error  then status = false ;      (*   error detected *)
   return  status , P, RE ;
End ; Version 6
```

5. COMPARISON OF CORRESPONDENT COMPUTING WITH RECOVERY BLOCKS AND N-VERSION PROGRAMMING

In this section we present a comparative study of Correspondent computing and the two well established schemes of Recovery Block and N-version programming. This study highlights some of the advantages as well as drawbacks of our proposed schemes.

5.1 CORRESPONDENT COMPUTING Vs. RECOVERY BLOCKS

In the recovery block scheme, each block consists of a primary, one or more alternates and an acceptance test. The primary process is executed upon entering the block. On completion the primary (or alternate) must pass an acceptance test to ensure the reliability of the process. If an error is detected, the system state is rolled back and an alternate is invoked. One of the drawbacks in recovery blocks is the difficulty in formulating error detection specifications for the acceptance test. The acceptance test in many cases is as complex as the program itself. This difficulty means that the test usually provides a means to check "reasonableness" rather than total "correctness". On the other hand, comparative testing in Correspondent computing, checks for correctness. The test is also very simple to specify, and it does not require extensive knowledge of any formal specifications and / or verification of assertions. Another advantage of the comparative test is that it can be performed at any point during execution of the program, unlike the acceptance test which can be performed only on exiting from the block. This feature allows errors to be detected dynamically during computation, the erroneous process may be aborted and an alternate invoked. If the error is detected during the early stages of computation a considerable amount of time is saved.

The sequential version of correspondent computing is slower than Recovery blocks in the best case, but faster in the worst case. The advantage of this scheme however is that it does not need recursive caches with all their associated overheads, nor does it need the acceptance test. Instead it relies on the Comparative test, and error recovery is in the forward direction rather than backward.

The concurrent version CS1 has all the advantages of SS over Recovery blocks and also overcomes the deficiencies of the sequential correspondent scheme in terms of time. Ofcourse to achieve the desired concurrency there is some extra overhead, but this is more than compensated for by the speed of concurrent execution.

The concurrent version CS2 has two apparent advantages over Recovery blocks. (1) Time efficiency : in case of an error being detected there is no roll back to try another alternate. (2) Space efficiency : since no roll back to a previous state occurs, there is no need to maintain recursive caches and their associated overheads. This scheme is faster in both the best and the worst case than the recovery block structure.

5.2 CORRESPONDENT COMPUTING Vs. N-VERSION PROGRAMMING

N-version programming relies on N (>= 2) different versions of the program , each version is independently designed from the same initial design specification, using various techniques, including different languages, algorithms, programmers etc. All N versions start together and execute concurrently. On completion of all the versions, a voting check is performed to determine the correct result. Synchronization of the versions is essential, as all versions start together, and the driver must wait for all versions to complete before a voting check can be performed. A timeout mechanism is incorporated to avoid unnecessary delays while waiting for an erroneous version to complete.

N-version programming is a powerful technique for fault tolerance, but it has some weaknesses. Redundancy in N-version programming is through the use of N different versions. It is at times difficult to provide versions that have redundancy of design. Correspondent computing, however, provides redundancy by using correspondent operations. These operations are generally easier to specify and implement, and quite possibly may be automatically generated in the future. The use of inexact voting check for error detection is another weakness of N-version programming. It is difficult to ensure complete correctness by using an inexact voting check, as it requires knowledge about the data, underlying machine, the tolerable range etc.

Comparing N-version programming with the sequential version SS of Correspondent computing, it is quite obvious that N-version programming is a better technique. N-version programming is more efficient in time-space, as it does not need to roll back to a previous state and try another alternative, or maintain recursive caches. The only advantage sequential Correspondent computing offers is its simple specification and redundancy requirements. As well as the ability to detect errors at any time during execution of the program.

However, the concurrent versions of Correspondent computing provide a challenge to N-version programming. CS1 executes concurrently and performs dynamic error detection. The ability to detect errors before completion of the program greatly improves the efficiency of CS1 and makes it faster than N-version programming. In N-version programming, the voting check has to wait for all versions to complete before it can test for errors, losing valuable computation time in continuing with an erroneous version. Another advantage of CS1 is since there are no separate versions, synchronization and time out mechanisms are not needed.

The second scheme CS2 has two or more versions executing concurrently. As each version has its own comparative test, the scheme is not constrained by the slowest version. The first version that returns the status = true is accepted and the other versions are ignored. Synchronization and timeout mechanisms are not needed as the versions donot have to wait for the others to complete.

5.3 EFFICIENCY ANALYSIS

This section presents a performance evaluation of our proposed scheme. And a comparison of their performance with that of Recovery blocks and N-version programming. Let us assume that each primary takes 20 units of time , Alternate 1 takes 25 units, and Alternate2 takes 30 units. Also assume that the each test takes 3 units of time. The correspondent operations are formed from the primary operation, therefore they use about the same amount of time as the primary i.e 20 units. Let us examine these schemes for the execution time of their best and worst case.

	Recovery Block	N-version prog.	SS	CS1	CS2
Worst Case	84	33	63	23	23
Best Case	23	33	63	23	23

Fig. 5 : Efficiency analysis results for Correspondent Computing Vs. Recovery Blocks and N-Version Programming.

Fig. 5 shows the time efficiency of the various schemes. It can be observed that only Recovery blocks diverge significantly in their best and worst case scenario, all the other schemes have constant time constraints for both the best and the worst case. Recovery blocks are considerably slower in the worst case, as they have to roll back their system state to an error free one and try the next alternate. Whereas, for the best case it completes in the shortest possible time, as it need only execute the primary. N-version programming is constrained by the slowest executing version, thus it has a constant time for the best and worst case. Both concurrent correspondent schemes have a higher efficiency in the worst case, as well as in the best case. Although, for the best case recovery blocks have the same time efficiency. The conclusion that can be ascertained from these observations is - Corresponding computing provides a faster and easier technique for fault tolerance.

6. CONCLUSION

In this paper, we have presented a new methodology for providing software fault tolerance. Correspondent computing provides a simple and elegant, yet powerful mechanism for redundancy. This useful redundancy is provided through the use of a correspondent operation for every primary operation. There are three types of correspondent operations,namely, reciprocal, duplicate and residual. It may not always possible to formulate all three operations for every primary operation. It is sufficient to use one or more correspondent operation. The degree of reliability would increase with the use of all three correspondent operations, but so would the space and time constraints. It is upto the user to select the best operation that suits the requirements, and provides efficient error detection and recovery. Error detection is provided through the use of a Comparative test. The specifications of the comparative test are based on the precise relationship between the primary and its correspondent operations.

Three different schemes have been suggested for implementing recovery, a sequential scheme, and two concurrent schemes. The selection of a particular scheme depends on the application under consideration, the system requirements, specification details, availability and reliability requirements,performance characteristics, time-space constraints and the development cost of the project.

Correspondent computing has the potential of becoming a powerful tool in the implementation of software fault tolerance. It is our desire to find out how easy (or difficult) it would be to automate the "generation" of correspondent operations, and the comparative test. Automatic generation would enable fault tolerance to be used in applications which otherwise would be very difficult to implement and / or to costly to develop . Our understanding of this new method would greatly benefit from at least one real implementation in a non-trivial system. These topics will be the main focus of our future research.

REFERENCES

[ANDE81] T. Anderson. and P. Lee, "Fault Tolerance principles and practice ", Prentice-Hall International, London, 1981.

[ANDE82] T. Anderson and P. Lee, "Fault tolerance terminology proposals ", 12th Annual International Conference on Fault Tolerant Computing, June 1982.

[CHEN78] L. Chen and A.Avizienis, "N-version programming : A fault-tolerance approach to reliability of software operation", 8th Annual International Conference on Fault Tolerant Computing, June 1978.

[LEE88] P.N. Lee, " Correspondent Computing", ACM Computer Science Conference, February 1988.

[RAND75] B.Randell, " System structure for software fault tolerance", IEEE trans. on Software Engg. Vol SE 1, No.2, June 1975.

IMPROVED UPPER BOUNDS FOR ALGORITHM-BASED FAULT TOLERANCE[†]

D.J.Rosenkrantz and S.S.Ravi
Department of Computer Science
SUNY at Albany
Albany, NY 12222

ABSTRACT

We consider the fault detection problem for multiple processor systems that use algorithm-based fault tolerance. For various parameterizations of the problem, we present schemes for constructing a test set that accomplishes fault detection. These schemes improve the known upper bounds on the number of tests required. For two faults, the number of tests used by our scheme matches the lower bound. For three faults, the number of tests used by our scheme is at most one more than the lower bound. We use this latter scheme in conjunction with a divide-and-conquer strategy to detect four or more faults. To detect $s \geq 4$ faults, the number of tests used by this strategy is within a factor of $O(\log_2 s)$ of the lower bound. We also discuss some special cases for which up to seven faults can be detected using a number of tests which is approximately equal to the lower bound for two faults.

1. INTRODUCTION

Algorithm-based fault tolerance [HA84] is a cost-effective approach by which results computed by the processors in a multiple processor system can be tested for errors caused by faulty hardware. In systems employing this approach, the processors operate on encoded input data and produce encoded outputs. The encoding of the outputs facilitates the use of on-line error checking procedures. The approach derives its name from the fact that the encoding method depends on the algorithm being executed by the processors. Algorithm-based fault tolerance has been used in conjunction with several types of algorithms, including matrix operations [LP88, HA84], evaluation of arithmetic expressions and polynomials [BA84], etc. Further discussion on this approach is available in [BA86a, BA86b, HA84].

An important problem to be addressed in this context is the design of the error checking procedures themselves with the goal of constructing test sets with a

---

[†] Research supported by NSF Grants DCR86-03184 and DCI86-03318 respectively.

minimum number of tests that accomplish fault detection. This issue was first addressed in [BA86a]. These researchers proposed a formal model for the problem and obtained upper and lower bounds on the number of tests needed for fault detection and fault location. In this paper, we study the fault detection problem and present new results that narrow the gap between the upper and lower bounds presented in [BA86a]. Algorithms for constructing the test set in a variety of situations are given. The constructed test sets are often much smaller than would be produced by currently known techniques. Due to space limitations, only sketches of proofs appear in this extended abstract. The interested reader is referred to [RR88] for details.

2. REVIEW OF THE BANERJEE-ABRAHAM MODEL

In order to detect hardware faults through on-line tests, it is necessary to assume that a hardware fault manifests itself as a discrepancy between the expected result of a computation and the observed result [BA86a]. The purpose of an on-line test is to detect such a discrepancy. Each output produced by the computing elements is considered a **data item**. A **test** is a hardware or a software procedure (or a combination thereof) applied on a subset of these data items. Each test generates a Boolean outcome (0 or 1). A test outputs a 0 if and only if *none* of the data items included in that test is in error. In practice, there usually are some limitations on the testing procedures, and those limitations create an issue of determining which data items should participate in each test. Often there is a limit on the number of data items that can be included in a test. Further, if the set of data items being operated upon by a given test includes too many faulty items, the outcome of the test may be *erroneous* (i.e., the test may output a 0 when it should output a 1). As shown in [BA86a], it is sufficient to analyze the case when *at most one* of the data items used in a test is in error. This is because tests that allow more than one faulty item can be obtained by combining tests that allow only one faulty item. In view of these considerations, we can now define the notion of a **valid test**[†]:

Definition 2.1: Given the value of the parameter g, a test is **valid** if and only if it satisfies the following two conditions:

(a) The test operates upon at most g data items.

(b) The test outputs a 0 if none of its data items is in error and outputs a 1 if *exactly one* of its data items is in error. If more than one of the data items is in error, the outcome of a test is unreliable.

Henceforth, we will use the term "test" to mean a "valid test". We are now in a position to define the notion of fault detection in a precise manner.

Definition 2.2: Let D be a collection of data items and C be a collection of tests on the data items in D. C **detects** s faults in D if and only if the following condition holds: for all j, $1 \leq j \leq s$, if D contains j faulty items, then *at least one test in* C *is guaranteed to output a 1.*

If the data set D does not contain any faulty items then each test in C will output a 0 since C contains only valid tests. We should also note that the set of tests C must be constructed in such a way that regardless of which subset F of D (with $1 \leq |F| \leq s$) is chosen as the set of faulty items, some test in C is guaranteed to output a 1. For simplicity, we will often use the phrase "test reports failure" to indicate that the test is guaranteed to output a 1, i.e. exactly one data item appearing in the test is faulty.

In practice, it is important to detect faults using as few tests as possible since the tests constitute an overhead on the computation. The authors of [BA86a] addressed this issue and obtained upper and lower bounds on the number of tests required. In the following section, we will present new fault detection schemes that lead to improved upper bounds.

[†] A valid test is a $(g,1)$ check in the terminology of [BA86a].

3. NEW SCHEMES FOR FAULT DETECTION

Throughout this paper, n will denote the number of data items in the set D and g will denote the upper bound on number of items that can be included in a test. We begin by noting that the problem of detecting 1 fault is trivial: $\lceil n/g \rceil$ tests are necessary and sufficient. So we need to consider only systems with 2 or more faults. For the case of $g = 2$, matching upper and lower bounds have been presented in [BA86a]. Therefore, we will assume throughout this paper that $g > 2$. To be consistent with the notation used in [BA86a], we use the symbol s to denote the number of faults and the symbol p to denote the number of tests in the set C used by a fault detection scheme.

3.1. A Detection Scheme for 2 Faults

It was shown in [BA86a] that at least $2n/(g+1)$ tests are necessary to detect 2 faults. This lower bound can be strengthened (slightly) to $\lceil 2n/(g+1) \rceil$ since the number of tests must be an integer. Our first result shows that $\lceil 2n/(g+1) \rceil$ tests are *sufficient*. This is a significant improvement over the upper bound of $gn/(2g-1)$ proven in [BA86a]. The following lemma plays a key role in the proof of this theorem as well as some of the results proved in the next section.

Lemma 3.1: Suppose a collection of tests C satisfies the following three conditions:

(a) Each data item appears in at least one test in C.
(b) For each pair c_1, c_2 of tests in C, *at most one* data item appears in both c_1 and c_2.
(c) Each test in C contains *at most one* data item that appears in just that test.

Then C is a valid scheme for detecting 2 faults.

Proof: If there is only one faulty item, then the faulty item appears in some test (by condition (a)) and that test reports failure. So, assume that there are two faulty items α and β. If α and β each appears in only one test, then they must occur in *different* tests (by condition (c)) and so both of these tests will report failure. Thus at least one of the objects, say α, appears in two or more tests. From property (b) above, β can occur in at most one of these tests. Therefore, there is at least one test containing α but not β and that test will report failure. □

Theorem 3.2: Suppose $s = 2$, $g > 2$ and $n \geq g^2-1$. Then, $p \leq \lceil 2n/(g+1) \rceil$.

Proof sketch: First, partition the set D of items (arbitrarily) into two subsets A and B such that $|B| = \lceil 2n/(g+1) \rceil$. With this choice for $|B|$, it can be shown that $|A|$ is at least $(g-1)^2$. To construct the set C of tests, envision the members of A as being arranged in a rectangular grid with $(g-1)$ columns, where the last row is possibly incomplete. Since A contains at least $(g-1)^2$ elements, there are at least $(g-1)$ complete rows.

The set C is composed of **row tests** and **column tests** constructed as follows:

(i) Each *row test* contains a row of the grid plus one element of B. If the last row is incomplete, say containing x elements where $x < (g-1)$, then the last row test contains the x elements of the last row, the first $(g-1-x)$ elements of the first column and a member[‡] of B.

(ii) Each *column test* (except possibly the last) contains $(g-1)$ adjacent elements of a column, with wraparound into the next column, plus one member of B. If the last row test includes some elements of the first column, then the column tests begin with the next element of the first column. The last column test contains the items of the last column which have not been used in any column test plus a member of B. Thus the last column test may use fewer than $(g-1)$ data items from A.

It can be shown that after the construction of the row and column tests, there is at most one element of B which has not appeared in any test. If B contains any such element, then it is placed in a separate test by itself.

[‡] Each member of B appears in *exactly* one test.

Observe that in the above test scheme, each element of A appears in *exactly* two tests. Using this observation, we can show that the number of tests does not exceed $|B|$ which is equal to $\lceil 2n/(g+1) \rceil$.

To prove that the above is a valid scheme for 2 faults, we show that the set C satisfies the conditions of Lemma 3.1. □

3.2. A Detection Scheme for 3 Faults

Clearly, $\lceil 2n/(g+1) \rceil$ is also a lower bound on the number of tests needed to detect 3 faults. The known upper bound for this case is $qn/(q+g-1)$ where $q = \lceil (3g+1)/2 \rceil$ [BA86a]. Our next theorem shows that this upper bound can be improved. The theorem uses the following lemma.

Lemma 3.3: Given a set of g^2 items, $2g-1$ tests are sufficient to detect 3 faults.

Proof sketch: Partition the g^2 elements (arbitrarily) into two sets A and B such that $|A| = g^2-g$ and $|B| = g$. Arrange the elements of A as a $g \times (g-1)$ grid. Construct g row tests, each containing a row of the grid plus a member of B. Construct $(g-1)$ column tests, each containing the g items in a column of A. This gives a total of $2g-1$ tests. Note that only row tests use an item from B. To prove that this scheme detects 2 faults, we appeal to Lemma 3.1. That the scheme detects 3 faults can be shown by a case analysis. □

If $n \le g^2$, we can conclude from the above lemma that $2g-1$ tests are sufficient. Therefore, it is more interesting to consider the upper bound when $n \ge g^2$.

Theorem 3.4: Suppose $s = 3$, $g > 2$ and $n \ge g^2$. Then,

$$p \le \lceil (2n - \lfloor n/g \rfloor)/g \rceil + 1$$

The construction of the test set to prove the above theorem uses ideas similar to those used in the proofs of Theorem 3.2 and Lemma 3.3. We refer the reader to [RR88] for details. In that paper, we also prove the following theorem.

Theorem 3.5: Suppose $s = 3$, $g > 2$ and $n \geq g^2$. Then, at least $\lceil (2n - \lfloor n/g \rfloor)/g \rceil$ tests are necessary to accomplish fault detection.

Theorem 3.5 indicates that our construction for detecting 3 faults uses *at most one* test more than the minimum required number.

3.3. A Scheme for 4 or more Faults

Our fault detection scheme for $s \geq 4$ uses Lemma 3.3 in combination with a divide-and-conquer strategy. The idea is illustrated in the following lemma.

Lemma 3.6: Let $s \geq 4$ and $r = \lceil \log_2(s/3) \rceil$. Given a set I containing g^{r+2} data items, $\frac{(r+2)\,|I|}{g} - g^r$ tests are sufficient to detect s faults in I.

Proof sketch: Partition the set I into g subsets $I_1, I_2, \ldots, I_g$ each containing g^{r+1} items. Assume for a moment that we have for each I_j $(1 \leq j \leq g)$, a set C_j of tests capable of detecting $\lfloor s/2 \rfloor$ faults in I_j. We can then construct a set C of tests to detect s faults in I as follows. C includes each C_j $(1 \leq j \leq g)$ and an additional set T of tests defined below. For $1 \leq j \leq g$, arrange the g^{r+1} items in I_j in any arbitrary order. T will contain g^{r+1} tests (numbered 1 through g^{r+1}) with the i^{th} test containing the item numbered i from each subset $I_1, I_2, \ldots, I_g$. It can be shown that C will indeed detect s faults in I.

The above construction is applied *recursively* to each I_j to construct C_j. The value of r has been so chosen that after the r^{th} stage, we will have subsets containing g^2 items and each subset is required to detect only 3 faults. We can then apply Lemma 3.3. □

We can now state our upper bound for detecting $s \geq 4$ faults.

Theorem 3.7: Suppose $s \geq 4$, $g > 2$ and $r = \lceil \log_2(s/3) \rceil$. Further, let $n > g^{r+2}$ and $k = \lceil n/g^{r+2} \rceil$. Then[†],

$$p \leq \left\lfloor \frac{(r+2)n}{g}(1 + \frac{1}{k-1}) - \frac{n}{g^2}(1 + \frac{1}{k}) \right\rfloor$$

Proof sketch: Partition the set D into subsets each containing g^{r+2} elements (the last subset may contain less). There are $\lceil n/g^{r+2} \rceil = k$ such subsets. For each subset we can use the construction presented in the proof of Lemma 3.6. Thus the total number of tests p is given by,

$$p \leq k\,((r+2)g^{r+1} - g^r)$$

Tedious but straightforward algebraic manipulations of the above expression yield the bound indicated in the statement of the theorem. □

Since $r = O(\log_2 s)$, the upper bound given by the above theorem is within a factor $O(\log_2 s)$ of the (trivial) lower bound of $\lceil 2n/(g+1) \rceil$. The known upper bound for $s \geq 3$ is $qn/(q+g-1)$ where $q = \lceil (sg+1)/2 \rceil$ [BA86a]. Our scheme uses fewer tests when $\log_2 s$ is small compared to g.

4. SOME SPECIAL CASES

In this section, we consider two special instances of the problem. For these instances, we prove that up to 5 and 7 faults (respectively) can be detected using $2n/g$ tests. Note that this number is very close to the lower bound of $\lceil 2n/(g+1) \rceil$ for detecting 2 faults. The design of testing schemes for these instances uses ideas from linear algebra, representations of integers in base g and modulo arithmetic.

Theorem 4.1: Suppose g is a *prime* and $n = kg^3$ for some positive integer k. Then there is a scheme with $2n/g$ tests which can detect 5 faults.

Proof sketch: Arrange the n elements in a rectangular grid with g columns. Divide the rows into blocks of g^2 rows each. The tests for each block are constructed as follows. Number the columns from 0 to $g-1$ and the rows from 0 to (g^2-1). Suppose that each row number is expressed as a 2 digit number in base g. Then each row number can be uniquely expressed as $ag+b$ where $0 \leq a,b < g$.

[†] Since $n > g^{r+2}$, k is at least 2. Thus $1/(k-1)$ is a well defined quantity.

For each row, there is a row test that contains the g elements of that row. For each column, there are g column tests obtained as follows. Assume that the column tests are numbered from 0 to $g-1$. For column i, the element in row $ag+b$ is included in test $a+ib$, where the arithmetic is done modulo g. We can show that the column tests contain disjoint sets of elements.

Each element is included in *exactly* two tests (one row test and one column test) and since each test includes exactly g elements, the number of tests is equal to $2n/g$.

The tests constructed as above satisfy the following "exclusion" property: If two elements from different rows appear in a column test, then no other column test will include elements from those two rows. The proof of this property uses the fact that since g is a prime, the set $\{0, 1, 2, ..., g-1\}$ with addition and multiplication modulo g forms a field. That the scheme detects 5 faults can be shown using the exclusion property and a case analysis. □

Theorem 4.2: Suppose $n = k g^{g+1}$ for some positive integer k. Then there is a scheme with $2n/g$ tests which can detect 7 faults.

Proof sketch: As before, arrange the objects in a grid with g columns. Consider a block g^g of rows. Number these rows from 0 to g^g-1 and express each row number in base g so that each row number consists of g digits.

For each row, there is a row test that contains the g elements of that row. For each column, there are g^{g-1} column tests. The column tests for column i are obtained by mapping the g digit row number into the $g-1$ digit number obtained by deleting the i^{th} digit from the row number.

As in the proof of Theorem 4.1, each item is involved in exactly two tests and so the number of tests is $2n/g$. We can show that the resulting set of tests satisfies the exclusion property mentioned above. That the scheme detects 7 faults can be shown through a case analysis. □

5. REFERENCES

[BA84] P.Banerjee and J.A.Abraham, "Fault-Secure Algorithms for Multiple Processor Systems", in *Proc. 11th International Symp. on Computer Architecture*, Ann Arbor, MI, June 1984, pp 279-287.

[BA86a] P.Banerjee and J.A.Abraham, "Bounds on Algorithm-Based Fault Tolerance in Multiple Processor Systems", IEEE Trans. Computers, Vol. C-35, No. 4, April 1986, pp 296-306.

[BA86b] P.Banerjee and J.A.Abraham, "Concurrent Fault Diagnosis in Multiple Processor Systems", in *Proc. 16th International Symp. on Fault Tolerant Computing (FTCS-16)*, July 1986, pp 298-303.

[HA84] K.H.Huang and J.A.Abraham, "Algorithm-Based Fault Tolerance for Matrix Operations", IEEE Trans. Computers, Vol. C-33, No. 6, June 1984, pp 518-528.

[LP88] F.T.Luk and H.Park, "An Analysis of Algorithm-Based Fault Tolerance Techniques", Journal of Parallel and Distributed Computing, Vol. 5, No. 2, April 1988, pp 172-184.

[RR88] D.J.Rosenkrantz and S.S.Ravi, "Improved Upper Bounds for Algorithm-Based Fault Tolerance", Tech. Report 88-22, Department of Computer Science, SUNY at Albany, June 1988.

ADAPTIVE DIAGNOSIS FOR PROBABILISTICALLY DIAGNOSABLE SYSTEMS

FENG-HSIEN WARREN SHIH
IBM Research Division
T. J. Watson Research Center
Yorktown Heights, NY 10598

KAZUO NAKAJIMA[†]
Electrical Engineering Department,
Institute for Advanced Computer Studies,
and Systems Research Center
University of Maryland
College Park, MD 20742

ABSTRACT

We show that the adaptive diagnosis approach can be applied to probabilistically diagnosable systems. Three adaptive diagnosis algorithms are developed under both the symmetric and asymmetric test invalidation assumptions. A test selection strategy based on a probabilistic measure of test results is derived and used in each algorithm. We show that the adaptive algorithms are efficient in identifying all faulty units in a system.

I. INTRODUCTION

Consider a system denoted by $S(U,T)$ or simply S, where $U = \{u_1, u_2, \ldots, u_n\}$ is a set of n units and T is a connection assignment of test links which assigns each unit to test a subset of the other units. Thus the system S can be represented by a digraph $G = (U,T)$, where U is the set of vertices of G and $T = \{(u_i, u_j) | u_i, u_j \in U$, and u_i is assigned to test $u_j\}$ is the set of directed edges of G. The result of each test is either "pass" (fault-free) or "fail" (faulty) and can be expressed as a binary weight $a(u_i, u_j)$ of edge (u_i, u_j) in T. In particular, $a(u_i, u_j) = 0$ if u_i judges u_j to be fault-free, and $a(u_i, u_j) = 1$ otherwise. The set of all test results of S is called the *syndrome* of S.

Preparata, Metze, and Chien [16] introduced the above graph theoretical model, which is now known as the *PMC model*, to deal with system-level faults in a multiprocessor system. In their model, it is assumed that the test result is reliable only if the testing unit is fault-free; otherwise it is unreliable regardless of the condition of the other unit involved in the test. This is called the *symmetric test invalidation* (abbreviated as STI) assumption. They defined a system S to be *t-diagnosable* if given a syndrome of S, all faulty units in S can correctly be identified, provided the number of faulty units does not exceed the given bound, t. A considerable amount of research has been done on the characterization, diagnosis and diagnosability of t-diagnosable systems under the PMC model [1,5,8,11,18].

[†] The work of this author was supported in part by National Science Foundation grants MIP-84-51510 and CDR-85-00108, and in part by a grant from AT&T.

A variation of the PMC model, called the *BGM model*, was later introduced by Barsi, Grandoni, and Maestrini [2]. In this model, besides the test invalidation assumption of the PMC model, it is further stipulated that a faulty unit is never evaluated to be fault-free. This is called the *asymmetric test invalidation* (abbreviated as ATI) assumption. A characterization theorem, diagnosis and diagnosability algorithms for t-diagnosable systems under the BGM model have been developed [2,13,15].

In all of the work mentioned above, it is implicitly assumed that all system units have the same probability of failure. In other words, all faults of the system are treated equally in the diagnosis process. On the other hand, Maheshwari and Hakimi [12] studied a probabilistic nature of the processor fault and proposed a probabilistic weighted model. In particular, they considered the *a priori* probability of failure associated with each unit in the system. They defined a system S to be *probabilistically* τ*-diagnosable*, or p-τ*-diagnosable* in short, if given a syndrome of S, the set of faulty units whose total *a priori* probability of failure is at least τ, can correctly be identified. Characterization theorems for this model under both the STI and ATI assumptions have been obtained [6,7,12]. An $O(n^3)$ diagnosis algorithm using network flow and vertex-cover set techniques has been developed [14]. Furthermore, the diagnosability problem for p-τ-diagnosable systems has been shown to be NP-complete [19].

In the conventional diagnosis approach discussed so far, one first chooses a set of diagnostic tests, then seeks the results of these tests, and finally proceeds to use the test results to identify the faulty units. Due to the high complexity of the diagnosis algorithms and the requirement for a complete syndrome, the on-line implementation of the diagnosis process becomes very difficult. To overcome this problem, Nakajima [14] suggested a so-called *adaptive diagnosis* approach. In this approach, tests are chosen and performed based on the results of the previous tests until one can identify a fault-free unit. This fault-free unit may then be used as a tester to identify all faulty units. Efficient diagnosis algorithms using this adaptive approach were later developed under both the PMC and BGM models [9]. Additional research work aiming at the reduction of the numbers of tests and testing rounds required for the diagnosis of a system in the PMC model has also been reported [3,10,17]. However, the application of the adaptive diagnosis approach to the more general p-τ-diagnosable systems has yet to be seen.

In this paper, we show for the first time that the adaptive diagnosis approach can be applied to probabilistically diagnosable systems. We develop three adaptive diagnosis algorithms for identifying a fault-free unit under both the STI and ATI assumptions. In the next section, we review the p-τ-diagnosable system model and obtain characterization theorems from previously known such theorems. We derive a test selection strategy based on a probabilistic measure of test results in Section III. In Section IV, we present the adaptive diagnosis algorithms based on the theorems given in Section II. The complexity analysis of each algorithm is also provided. Section V concludes the paper.

II. PRELIMINARIES

Let $S(U,T)$ be a system, where $U = \{u_1, u_2, \ldots, u_n\}$ and $T = \{(u_i, u_j) | u_i, u_j \in U$, and u_i is assigned to test $u_j\}$.

Definition 1: A subset $F \subset U$ is called an *allowable fault set* of S if given a syndrome of S, (1) $u_i \in U - F$ and $a(u_i, u_j) = 0$ imply $u_j \in U - F$ and (2) $u_i \in U - F$ and $a(u_i, u_j) = 1$ imply $u_j \in F$. ☐

Thus, F is an allowable fault set if the assumptions that the units contained in F are faulty and that the units contained in $U - F$ are fault-free are consistent with the given syndrome.

Let $p(u_i)$ be the *a priori* probability of failure of unit u_i. Assuming that all $p(u_i)$'s are independent, the *a priori* probability of the set of faulty units $F \subseteq U$ in S is given as

$$P(F) = \prod_{u_i \in F} p(u_i) \prod_{u_i \in U-F} (1 - p(u_i)) \, .$$

Definition 2: A system S is called *probabilistically τ-diagnosable* or simply *p-τ-diagnosable* if for any syndrome of S, there exists a unique allowable fault set $F \subseteq U$ such that $P(F) \geq \tau$, assuming that the *a priori* probability of occurrence of the set of faulty units in S is at least τ. ☐

Let $F \subseteq U$ be a set of faulty units such that

$$P(F) = \prod_{u_i \in F} p(u_i) \prod_{u_i \in U-F} (1 - p(u_i)) \geq \tau.$$

Then

$$\log\,(P(F)) = \sum_{u_i \in F} \log\,(p(u_i)) + \sum_{u_i \in U-F} \log\,(1 - p(u_i)) \geq \log\,(\tau) \, .$$

It follows that

$$\sum_{u_i \in F} \log\left(\frac{1 - p(u_i)}{p(u_i)}\right) \leq -\log\,(\tau) + \sum_{u_i \in U} \log\,(1 - p(u_i)) \triangleq K(\tau) \, .$$

If with each unit $u_i \in U$ a weight $w(u_i) = \log\,[(1 - p(u_i))/p(u_i)]$ is associated, then the problem of identifying the set of faulty units in a p-τ-diagnosable system S is equivalent to that of determining the allowable fault set F for which

$$W(F) = \sum_{u_i \in F} w(u_i) \leq K(\tau) \, .$$

The class of conventional t-diagnosable systems then becomes a special case of p-τ-diagnosable systems in which the weight of each unit is identical. Note that in the above model, it is assumed that $p(u_i) < 1/2$ and hence $w(u_i) > 0$ for every $u_i \in U$.

Under the STI assumption, the test result $a(u_i, u_j)$ is reliable only if u_i is a fault-free unit. On the other hand, under the ATI assumption, unit u_j is fault-free if $a(u_i, u_j) = 0$ regardless of the condition of unit u_i. The following theorems give necessary and sufficient conditions for a system to be p-τ-diagnosable under both assumptions. They are derived directly from those given by Fujiwara and Kinoshita [6,7]. Let $S(U, T)$ be a system with each unit $u_i \in U$ having weight $w(u_i)$, where its corresponding digraph $G = (U, T)$ is assumed to be fully connected.

Theorem 1: S is p-τ-diagnosable under the STI assumption if and only if there exists no 2-partition (V_1, V_2) of U such that $W(V_1) \leq K(\tau)$ and $W(V_2) \leq K(\tau)$. □

Theorem 2: S is p-τ-diagnosable under the ATI assumption if and only if there exists no 3-partition $(V, \{u\}, \{v\})$ of U such that $W(V) + w(u) \leq K(\tau)$ and $W(V) + w(v) \leq K(\tau)$. □

III. TEST SELECTION STRATEGY

In order to achieve efficient diagnosis using the adaptive approach, the selection of tests among various units in a system is very important in all diagnosis algorithms [3,9,10]. For t-diagnosable systems, the performance of such an algorithm is simply judged by the number of tests used in the diagnosis, since the probability of failure is assumed to be equal for each unit in the system. In this case, when selecting a new unit in the adaptive diagnosis process, only the connection assignment of that unit has to be considered. However, for p-τ-diagnosable systems, the performance of an algorithm should also be judged by the probability of the occurrence of an adaptive test sequence. This in turn requires the consideration of the probability of failure for each unit in the test selection strategy.

Let $S(U, T)$ be a system with each unit $u_i \in U$ having *a priori* probability of failure $p(u_i)$. Let $P_0(u_i, u_j)$ denotes the probability that the test result $a(u_i, u_j) = 0$.

Under the STI assumption, it is assumed that (1) $a(u_i, u_j) = 0$ if both u_i and u_j are fault-free, and (2) $a(u_i, u_j) = 0$ or 1 if u_i is faulty. A natural interpretation of the second case is that $P_0(u_i, u_j) = 1/2$ if u_i is faulty. In this paper, however, we treat this case in a more general setting, by introducing two probabilistic measures. Let α (resp., β) be the probability of the occurrence of $a(u_i, u_j) = 0$ when u_i is faulty and u_j is fault-free (resp., faulty). Note that $0 \leq \alpha$, $\beta \leq 1$. Furthermore, it is reasonable to stipulate that $\alpha + \beta \leq 1$ since if the testing unit u_i is faulty, it is more likely to have $a(u_i, u_j) = 1$ than $a(u_i, u_j) = 0$. Based on these assumptions, $P_0(u_i, u_j)$ can be expressed as

$$P_0(u_i, u_j) = (1 - p(u_i))(1 - p(u_j)) + \alpha p(u_i)(1 - p(u_j)) + \beta p(u_i)p(u_j) .$$

smallskip
Under the ATI assumption, (1) $a(u_i, u_j) = 0$ if both u_i and u_j are fault-free, (2) $a(u_i, u_j) = 1$ if both u_i and u_j are faulty, and (3) $a(u_i, u_j) = 0$ or 1 if u_i is faulty and u_j is fault-free. Using the same probabilistic measures α and β introduced above, the second case implies that $\beta = 0$ while the third case can be treated with α in the same way as under the STI assumption. Thus, we have

$$P_0(u_i, u_j) = (1 - p(u_i))(1 - p(u_j)) + \alpha p(u_i)(1 - p(u_j)) .$$

Theorem 3: For any two units $u_i, u_j \in U$, if $p(u_i) \leq p(u_j)$, then $P_0(u_j, u_i) \geq P_0(u_i, u_j)$ under both the STI and ATI assumptions.

Proof: Under the STI assumption, we have

$$P_0(u_j, u_i) = (1 - p(u_j))(1 - p(u_i)) + \alpha p(u_j)(1 - p(u_i)) + \beta p(u_j)p(u_i) \text{ , and}$$
$$P_0(u_i, u_j) = (1 - p(u_i))(1 - p(u_j)) + \alpha p(u_i)(1 - p(u_j)) + \beta p(u_i)p(u_j) .$$

It follows that

$$\begin{aligned} P_0(u_j,u_i) - P_0(u_i,u_j) &= \alpha p(u_j)(1-p(u_i)) - \alpha p(u_i)(1-p(u_j)) \\ &= \alpha(p(u_j) - p(u_i)) \geq 0 \ . \end{aligned}$$

Under the ATI assumption, we have

$$\begin{aligned} P_0(u_j,u_i) &= (1-p(u_j))(1-p(u_i)) + \alpha p(u_j)(1-p(u_i)) \ , \text{ and} \\ P_0(u_i,u_j) &= (1-p(u_i))(1-p(u_j)) + \alpha p(u_i)(1-p(u_j)) \ . \end{aligned}$$

Hence,

$$P_0(u_j,u_i) - P_0(u_i,u_j) = \alpha(p(u_j) - p(u_i)) \geq 0 \ . \ \square$$

The main idea of the adaptive diagnosis approach is to identify a fault-free unit as quickly as possible. Under the ATI assumption, once a "0" test result is obtained, the unit tested is fault-free. Under the STI assumption, the accumulation of "0" test results as well as that of "1" test results will lead to the identification of a fault-free unit. Theorem 3 implies that in order to produce a "0" test result probabilistically quicker, a unit with a higher probability of failure should test a unit with a lower probability. Without loss of generality, we can assume, in the remainder of the paper, that the units in U are sorted in nondecreasing order of their *a priori* probabilities of failure; that is, $p(u_1) \leq p(u_2) \leq \ldots \leq p(u_n)$, or equivalently, $w(u_1) \geq w(u_2) \geq \ldots \geq w(u_n)$.

Theorem 4: For any three units $u_i, u_j, u_k \in U$, if $p(u_i) \leq p(u_j) \leq p(u_k)$, then $P_0(u_j,u_i) \geq P_0(u_k,u_i) \geq P_0(u_k,u_j)$ under both the STI and ATI assumptions.

Proof: Under the STI assumption, we have

$$\begin{aligned} P_0(u_j,u_i) &= (1-p(u_j))(1-p(u_i)) + \alpha p(u_j)(1-p(u_i)) + \beta p(u_j)p(u_i) \ , \\ P_0(u_k,u_i) &= (1-p(u_k))(1-p(u_i)) + \alpha p(u_k)(1-p(u_i)) + \beta p(u_k)p(u_i) \ , \text{ and} \\ P_0(u_k,u_j) &= (1-p(u_k))(1-p(u_j)) + \alpha p(u_k)(1-p(u_j)) + \beta p(u_k)p(u_j) \ . \end{aligned}$$

Therefore, we derive

$$\begin{aligned} P_0(u_j,u_i) - P_0(u_k,u_i) &= (p(u_k) - p(u_j))(1-p(u_i)) + \alpha(p(u_j) - p(u_k))(1-p(u_i)) \\ &\quad + \beta(p(u_j) - p(u_k))p(u_i) \\ &= (p(u_k) - p(u_j))[1 - p(u_i) - \alpha(1-p(u_i)) - \beta p(u_i)] \\ &= (p(u_k) - p(u_j))[1 - \alpha - (1-\alpha+\beta)p(u_i)] \\ &\geq (p(u_k) - p(u_j))[1 - \alpha - (1-\alpha+\beta)\cdot\frac{1}{2}](\text{since } p(u_i) < \frac{1}{2}) \\ &= (p(u_k) - p(u_j))(\frac{1}{2})(1-\alpha-\beta) \geq 0(\text{since } \alpha+\beta \leq 1), \text{ and} \end{aligned}$$

$$\begin{aligned} P_0(u_k,u_i) - P_0(u_k,u_j) &= (1-p(u_k))(p(u_j) - p(u_i)) + \alpha p(u_k)(p(u_j) - p(u_i)) \\ &\quad + \beta p(u_k)(p(u_i) - p(u_j)) \ . \\ &= (p(u_j) - p(u_i))(1 - p(u_k) + \alpha p(u_k) - \beta p(u_k)) \\ &= (p(u_j) - p(u_i))[1 - (1-\alpha+\beta)p(u_k)] \geq 0 \\ &\quad (\text{since } p(u_k) < \frac{1}{2} \text{ and } 1+\beta-\alpha \leq 2) \ . \end{aligned}$$

Under the ATI assumption, we have

$$P_0(u_j,u_i) = (1-p(u_j))(1-p(u_i)) + \alpha p(u_j)(1-p(u_i))$$
$$P_0(u_k,u_i) = (1-p(u_k))(1-p(u_i)) + \alpha p(u_k)(1-p(u_i)) \text{ , and}$$
$$P_0(u_k,u_j) = (1-p(u_k))(1-p(u_j)) + \alpha p(u_k)(1-p(u_j)) \text{ .}$$

Therefore, we obtain

$$\begin{aligned} P_0(u_j,u_i) - P_0(u_k,u_i) &= (1-p(u_i))(p(u_k)-p(u_j)) + \alpha(1-p(u_i))(p(u_j)-p(u_k)) \\ &= (1-\alpha)(1-p(u_i))(p(u_k)-p(u_j)) \geq 0 \text{ , and} \\ P_0(u_k,u_i) - P_0(u_k,u_j) &= (1-p(u_k))(p(u_j)-p(u_i)) + \alpha p(u_k)(p(u_j)-p(u_i)) \\ &= (1-(1-\alpha)p(u_k))(p(u_j)-p(u_i)) \geq 0 \text{ . } \square \end{aligned}$$

The above theorems suggest the following test selection strategy. First, use tests of the form (u_{i+1}, u_i) rather than those of the form (u_i, u_{i+1}). Then, perform the test (u_2, u_1) rather than the test (u_3, u_2), and proceed to carry out the tests $(u_3, u_1), (u_4, u_1), \ldots$, until unit u_1 is hopefully identified to be fault-free; otherwise repeat the process starting with the test (u_3, u_2) and so forth. Based on this strategy, we develop adaptive diagnosis algorithms in the next section.

IV. ADAPTIVE DIAGNOSIS ALGORITHMS

Let $S(U,T)$ be a system with each unit $u_i \in U$ having weight $w(u_i)$. Recall the assumption that the units in U are sorted in nondecreasing order of their *a priori* probabilities of failure, or equivalently in nonincreasing order of their weights; that is, $w(u_1) \geq w(u_2) \geq \ldots \geq w(u_n)$. Before we proceed to show the adaptive diagnosis algorithms, we introduce optimal base sets for p-τ-diagnosable systems under the STI and ATI assumptions. They play a key role in the algorithms. Let $G = (U,T)$ be the digraph representing system S. We say that a subset $V \subseteq U$ of S is p-τ-diagnosable if the subsystem of S defined by the induced subgraph $(V, T(V))$ of G is p-τ-diagnosable, where $T(V) = \{(u,v) \in T | u, v \in V\}$. Namely, the subsystem consists of the units in V and the test links of T which connect units in V.

Definition 3: Let $S(U,T)$ be a p-τ-diagnosable system under the STI (resp., ATI) assumption. A smallest cardinality subset $B \subseteq U$ of S that is p-τ-diagnosable under the STI (resp., ATI) assumption, is called an *optimal base set* of S. $\square$

Given a p-τ-diagnosable system $S(U,T)$, one can construct an optimal base set B_s under the STI assumption in the following manner: if $\sum_{i=1}^{n} w(u_i) \leq 2K(\tau)$, set $B_s = U$; otherwise, find index b such that $\sum_{i=1}^{b-1} w(u_i) \leq 2K(\tau) < \sum_{i=1}^{b} w(u_i)$, and set $B_s = \{u_1, u_2, \ldots, u_b\}$. Similarly, an optimal base set B_a under the ATI assumption can be obtained as follows: if $\sum_{i=2}^{n} w(u_i) \leq K(\tau)$, set $B_a = U$; otherwise, find index b such that $\sum_{i=3}^{b-1} w(u_i) \leq K(\tau) - w(u_1) < \sum_{i=3}^{b} w(u_i)$, and set $B_a = \{u_1, u_2, \ldots, u_b\}$.

We now show an adaptive diagnosis algorithm which identifies a fault-free unit under the STI assumption. Let $B_s = \{u_1, u_2, \ldots, u_b\}$ be an optimal base set of S.

Algorithm SYM:

```
    $i \leftarrow 1$; $F \leftarrow \phi$; $k \leftarrow K(\tau)$
    while $i \leq b$ do
        if $W(F) + w(u_i) > K(\tau)$ then return */$u_i$ is fault-free /*
        else $V_0 \leftarrow \{u_i\}$; $V_1 \leftarrow \phi$; $j \leftarrow i + 1$
            while $j \leq b$ do
                Perform test $(u_j, u_i)$
                if $a(u_j, u_i) = 0$ then $V_0 \leftarrow V_0 \cup \{u_j\}$
                    if $W(V_0) > k$ then return */$u_i$ is fault-free /*
                else $V_1 \leftarrow V_1 \cup \{u_j\}$
                    if $W(V_1) > k$ then $F \leftarrow F \cup \{u_i\}$; $k \leftarrow k - w(u_i)$; $i \leftarrow i + 1$; break */$u_i$ is faulty /*
                    if $W(F) + w(u_i) > K(\tau)$ then return */$u_i$ is fault-free /*
                $j \leftarrow j + 1$
            endwhile
        $i \leftarrow i + 1$
    endwhile□
```

Let $\lceil x \rceil$ denote the smallest integer that is greater than or equal to x.

Theorem 5: Let $S(U, T)$ be a system which satisfies the condition of Theorem 1, and let B_s be an optimal base set of S with $|B_s| = b$. Under the STI assumption, Algorithm SYM always identifies a fault-free unit in S using at most $bx - x(x+1)/2$ tests, where $x = \lceil (b-1)/2 \rceil$.

Proof: We first consider the trivial case in which the algorithm terminates on the third line. In this case $F = \phi$ and $w(u_1) > K(\tau)$. Apparently u_1 cannot belong to any allowable fault set of cardinality at most $K(\tau)$, and hence u_1 is fault-free.

Next we consider the other two cases in which the algorithm terminates. Note first that the set V_1 contains those units that are faulty if unit u_1 is assumed to be fault-free. Therefore, if $W(V_1) > k$, V_1 cannot be a set of faulty units, and hence the assumption that u_1 is fault-free is wrong. Thus, u_1 is faulty. This implies that F always contains faulty units which have correctly been identified so far. Therefore, if the termination condition that $W(F) + w(u_i) > K(\tau)$ is met, u_i must be fault-free. On the other hand, the set V_0 contains those units that are faulty if unit u_1 is assumed to be faulty. Using the same reasoning as above, if the algorithm terminates when $W(V_0) > k$, u_i is fault-free.

In all of the above cases, the algorithm correctly identifies a fault-free unit if it terminates. We now show that it does terminate using at most $bx - x(x+1)/2$ tests. Recall that the units in the optimal base set B_s are sorted in nonincreasing order of their weights, that is, $w(u_1) \geq w(u_2) \geq \ldots \geq w(u_b)$. Let $V_x = \{u_i | 1 \leq i \leq x\}$ and $V_y = U - V_x - \{u_{x+1}\}$, where $x = \lceil (b-1)/2 \rceil$. Clearly $W(V_x) \geq W(V_y)$ and hence $W(V_x) + w(u_{x+1}) > K(\tau)$ since $\sum_{i=1}^{b} w(u_i) > 2K(\tau)$ due to the property of B_s. Therefore, in the worst case in which no unit in V_x is identified as fault-free in the diagnosis process, u_{x+1} will be diagnosed as fault-free. Clearly, the total number of tests used is at most $(b-1) + (b-2) + \ldots + (b-x) = bx - x(x+1)/2$. □

We now present two adaptive diagnosis algorithms which identify a fault-free unit under the ATI assumption. The first algorithm is applicable to systems which satisfy the condition of Theorem 1. Note that if a system is p-τ-diagnosable under the STI assumption, it is so under the ATI assumption. Let $B_s = \{u_1, u_2, \ldots, u_b\}$ be an optimal base set of S.

Algorithm ASYM1:

```
    $i \leftarrow 2;\ F \leftarrow \phi$
    while $i \le b$ do
        if $W(F) + w(u_1) > K(\tau)$ then return */$u_1$ is fault-free /*
        else Perform test $(u_{i+1}, u_i)$
            if $a(u_{i+1}, u_i) = 0$ then return */$u_i$ is fault-free /*
            else $F \leftarrow F \cup \{u_{i+1}\}$
        $i \leftarrow i + 2$
    endwhile
    print "$u_1$ is fault-free"□
```

Theorem 6: Let $S(U,T)$ be a system which satisfies the condition of Theorem 1, and let B_s be an optimal base set of S with $|B_s| = b$. Under the ATI assumption, Algorithm ASYM1 always identifies a fault-free unit in S using at most $\lceil (b-1)/2 \rceil$ tests.

Proof: Let $V_1 = \{u_i \in B_s | i \text{ is odd and } i \ge 3\}$ and $V_2 = \{u_i \in B_s | i \text{ is even }\}$. Since $w(u_{2j-1}) \ge w(u_{2j})$ for $j = 1, 2, \ldots, \lceil (b-1)/2 \rceil$, $w(u_1) + W(V_1) \ge W(V_2)$. Due to the property of the optimal base set B_s under the STI assumption and Theorem 1, $w(u_1) + W(V_1) > K(\tau)$. In the worst case, the results of all tests performed are "1". In this case, $F = V_1$ and hence $w(u_1) + W(F) > K(\tau)$. Thus, the algorithm terminates and u_1 is a fault-free unit. Clearly, the total number of tests used is at most $\lceil (b-1)/2 \rceil$. □

The last algorithm we present is applicable to systems which satisfy the condition of Theorem 2. Let $B_a = \{u_1, u_2, \ldots, u_b\}$ be an optimal base set of S.

Algorithm ASYM2:

```
    $i \leftarrow 2;\ j \leftarrow 2;\ F \leftarrow \phi;$
    while $i \le b$ do
        if $w(u_1) + W(F) > K(\tau)$ then return */$u_1$ is fault-free /*
        else Perform test $(u_i, u_1)$
            if $a(u_i, u_1) = 0$ then return */$u_1$ is fault-free /*
            else $j \leftarrow i + 1$
            while $j \le b$ do
                Perform test $(u_j, u_i)$
                if $a(u_j, u_i) = 0$ then return */$u_i$ is fault-free /*
                $j \leftarrow j + 1$
            endwhile
        $F \leftarrow F \cup \{u_i\}$
        $i \leftarrow i + 1$
    endwhile□
```

Theorem 7: Let $S(U,T)$ be a system which satisfies the condition of Theorem 2, and let B_a be an optimal base set of S with $|B_a| = b$. Under the ATI assumption, Algorithm ASYM2 always identifies a fault-free unit using at most $(b-1)b/2 - 1$ tests.

Proof: Let $V = \{u_i | 3 \leq i \leq b\}$. Due to the property of the optimal base set B_a under the ATI assumption, $w(u_1) + W(V) > K(\tau)$. During the first cycle, if the results of all tests that involve unit u_2 are "1", u_2 is faulty; otherwise, all units in $\{u_1\} \cup V$ would be faulty and their total weight would exceed $K(\tau)$. Using a similar reasoning, during the i-th cycle, if the results of all tests that involve unit u_{i+1} are "1", u_{i+1} is faulty. If no fault-free unit is found in $V - \{u_b\}$, $F = \{u_2\} \cup V - \{u_b\}$ and $w(u_1) + W(F) > K(\tau)$. Thus, the algorithm terminates and unit u_1 is fault-free. Therefore, the algorithm always identifies a fault-free unit. Clearly, the total number of tests used is at most $(b-1) + (b-2) + \ldots + 2 = (b-1)b/2 - 1$. □

V. CONCLUSIONS

We have presented efficient adaptive diagnosis algorithms for probabilistically diagnosable systems under both the symmetric and asymmetric test invalidation assumptions. We have given the complexity analysis of each algorithm for identifying a fault-free unit. We have derived a test selection strategy using the property of an optimal base set so as to achieve a higher probability of termination of the algorithms using the fewest tests. The adaptive approach has been shown to provide simple and efficient fault diagnosis even for the more general system-level model.

REFERENCES

[1] F. J. Allan, T. Kameda, and S. Toida, "An approach to the diagnosability analysis of a system," *IEEE Trans. Comput.*, vol. C-24, pp. 1040-1042, Oct. 1975.

[2] F. Barsi, F. Grandoni, and P. Maestrini, "A theory of diagnosability of digital systems," *IEEE Trans. Comput.*, vol. C-25, pp. 585-593, June 1976.

[3] M. Blecher, "On a logical problem," *Discrete Math.*, vol. 3, pp. 379-380, 1982.

[4] A. T. Dahbura, "An efficient algorithm for identifying the most likely fault set in a probabilistically diagnosable system," *IEEE Trans. Comput.*, vol. C-36, pp. 354-356, Apr. 1986.

[5] A. T. Dahbura and G. M. Masson, "An $O(n^{2.5})$ fault identification algorithm for diagnosable systems," *IEEE Trans. Comput.*, vol. C-33, pp. 486-492, June 1984.

[6] H. Fujiwara and K. Kinoshita, "Connection assignments for probabilistically diagnosable systems," *IEEE Trans. Comput.*, vol. C-27, pp. 280-283, Mar. 1978.

[7] H. Fujiwara and K. Kinoshita, "Some existence theorems for probabilistically diagnosable systems," *IEEE Trans. Comput.*, vol. C-27, pp. 379-384, Apr. 1978.

[8] S. L. Hakimi and A. T. Amin, "Characterization of connection assignment of diagnosable systems," *IEEE Trans. Comput.*, vol. C-23, pp. 86-88, Jan. 1974.

[9] S. L. Hakimi and K. Nakajima, "On adaptive system diagnosis," *IEEE Trans. Comput.*, vol. C-33, pp. 234-240, Mar. 1984.

[10] S. L. Hakimi and E. F. Schmeichel, "An adaptive algorithm for system level diagnosis," *J. Algorithms*, pp. 526-530, 1984.

[11] T. Kameda, S. Toida, and F. J. Allan, "A diagnosis algorithm for networks," *Inform. and Contr.*, vol. 29, pp. 141-148, 1975.

[12] S. N. Maheshwari and S. L. Hakimi, "On models for diagnosable systems and probabilistic fault diagnosis," *IEEE Trans. Comput.*, vol. C-25, pp. 228-236, Mar. 1976.

[13] G. G. L. Meyer, "A diagnosis algorithm for the BGM system level fault model," *IEEE Trans. Comput.*, vol. C-33, pp. 756-758, Aug. 1984.

[14] K. Nakajima, "A new approach to system diagnosis," in *Proc. 19th Annu. Allerton Conf. Commun. Contr. and Comput.*, Sept. 1981, pp. 697-706.

[15] J. Narasimhan and K. Nakajima, "An algorithm for determining the fault diagnosability of a system," *IEEE Trans. Comput.*, vol. C-35, pp. 1004-1008, Nov. 1986.

[16] F. P. Preparata, G. Metze, and R. T. Chien, "On the connection assignment problem of diagnosable systems," *IEEE Trans. Electron. Comput.*, vol. EC-16, pp. 848-854, Dec. 1967.

[17] E. F. Schmeichel, S. L. Hakimi, M. Ohtsuka, and G. Sullivan, "Minimizing the Number of Test Rounds," in *Dig. 18th Int. Symp. on Fault-Tolerant Computing Symp.*, Tokyo, Japan, July 1988, pp. 266-271.

[18] G. Sullivan, "A polynomial time algorithm for fault diagnosability," in *Proc. 25th Annu. Symp. Found. of Comp. Sci.*, Oct. 1984, pp. 148-155.

[19] G. Sullivan, "System-level fault diagnosability in probabilistic and weighted models," in *Dig. 17th Int. Symp. on Fault-Tolerant Computing*, Pittsburgh, PA, June 1987, pp. 190-195.

PROPERTIES OF c-CORRECTABILITY IN SELF-DIAGNOSING SYSTEMS

JON T. BUTLER*
Department of Electrical
and Computer Engineering
Naval Postgraduate School, Code 62-Bu
Monterey, CA 93943-5004

YONINA ROSEN
Department of Electrical
Engineering
University of Delaware
Newark, DE 19716

ABSTRACT

We propose a distributed disabling algorithm for a multiprocessing system in which each processor or unit is prevented from doing computation when it fails some number of tests by other units. The goal is to disable all faulty units and to enable all fault-free units. Specifically, a unit is disabled iff it fails d or more tests by enabled units (d-disabling rule). A multiprocessor system is c-correctable using the d-disabling rule iff all faulty units are permanently disabled and all fault-free units are permanently enabled after a finite number of applications of the disabling rule, provided there are no more than c faulty units. This models an unattended system where the removal of faulty units is done locally by simple and reliable circuitry. We give a sufficient condition for c-correctability in general systems and a necessary and sufficient condition in general systems where $c < d$. Then, we give necessary and sufficient conditions for c-correctability of two types of systems, (1) complete digraphs and (2) a new class of systems called segmented systems.

I. INTRODUCTION

In the systems diagnosis approach to reliability, testing is distributed. For example, in a multiprocessing system, processors test other processors producing pass or fail test results. The goal is to identify faulty units in the presence of incorrect information from such units. If there are too many faulty units, it may be impossible to uniquely identify them. For example, if *all* units are faulty, they may all produce pass test results, and it is impossible to distinguish between this and the case where all units are fault-free.

While testing is distributed, diagnosis may not be. Most papers on this subject have assumed a central diagnoser. In this case, system reliability depends critically on the reliability of the diagnoser. There has been a trend in recent years towards

---

*Research supported by NSF Grant #ECS-8203276 and a NAVELEX Chair Professorship tenured at the Naval Postgraduate School.

systems where the diagnosis is also distributed [2-8]. Meyer and Masson [7] propose a distributed diagnosis algorithm in which each unit has a "view" of the entire system based on tests it makes and on test results received by units that it finds to be fault-free. It is shown that, if there is an upper limit on the number of faulty units, the most common "view" is the correct one. This model is extended by Kuhl and Reddy [5,6] and Hosseini, Kuhl, and Reddy [3] to the case where links between units can also fail. Both are is based on the Preparata, Metze, and Chien [9] model of systems diagnosis. However, there is then the problem of how the user identifies faulty units and removes them from the system. Kreutzer and Hakimi [4] address the first problem but not he second. A distributed diagnosis algorithm based on the Russell and Kime [10] model is shown by Holt and Smith [2]. Repair and graceful degradation models are proposed, using a message passing method in which fault-free units try to gain an accurate view of the status of various other units.

The problem of reliably disabling faulty units in systems diagnosis has received little attention. To the credit of Holt and Smith [2], "controllers" are proposed that disable units diagnosed as faulty. Unlike previous papers, we consider self-diagnosis in which the process of disabling faulty units is inherent. That is, the process of disabling a unit is built-in to the diagnosis algorithm. The reliable operation of the system depends on the reliability of a circuit which implements the rule. We choose to make the function of this circuit so simple that ultrareliability is achieved inexpensively (by redundancy, for example). Specifically, a unit is disabled iff it fails d tests by enabled units. This is the d-disabling rule. We assume an upper bound c on the number of faulty units, and we seek conditions which guarantee that all faulty units are disabled and all fault-free units are enabled after a finite number of applications of the d-disabling rule. A sufficient condition for c-correctability is given for general systems. The condition is expressed as a property of subsets of units and how they are interconnected by tests. A necessary and sufficient condition for c-correctability using the d-disabling rule is given for general systems in which $d < c$ holds. Next, we show necessary and sufficient conditions for two specific classes of systems

1. complete digraphs and
2. segmented systems.

The latter systems are new. They have a cyclical symmetry that extends over groups of units.

This paper is arranged as follows. Section III shows a sufficient condition for c-correctability in general systems. Section IV gives necessary and sufficient conditions for c-correctability in two specific systems.

II. BACKGROUND AND NOTATION

A *system* is a directed graph where nodes represent units or processors and arcs represent tests between units. Let $V = \{u_0, u_1, \cdots, u_{n-1}\}$ be the set of units in the system. Then, a directed arc exists from u_i to u_j iff u_i tests u_j. The test outcome is either pass or fail, depending on the status of the units involved in the test. Each unit is either *fault-free* or *faulty*. If the testing unit is fault-free, then the test outcome is a true representation of the status of the tested unit, pass if the tested unit is fault-free and fail if it is faulty. However, if the testing unit is faulty, the test outcome is arbitrarily pass or fail.

A complete set of test results is called a *syndrome*. The object of a diagnosis is to identify uniquely all faulty units given a syndrome. If the number of faulty units is small enough, then unique identification is possible for *all* possible arrangements of faulty units and *all* possible syndromes. Specifically, a system is *t-diagnosable* iff all faulty units can be uniquely identified provided there are no more than t of them. Preparata, Metze, and Chien [9] show necessary conditions for a system to be t-diagnosable and Hakimi and Amin [1] show necessary and sufficient conditions.

Each unit is either *enabled* or *disabled.* We assume initially that any unit can be arbitrarily enabled or disabled.

Definition: The *d-disabling rule* is as follows: a unit is disabled if it fails d or more tests by enabled units; otherwise, it is enabled.

The rule is applied continually to each unit without regard to order among units. We seek conditions that guarantee a faulty unit is eventually disabled and remains disabled at each application of the d-disabling rule and that a fault-free unit is similarly enabled.

Definition: A system is *c-correctable* using the d-disabling rule iff for

1. any arrangement of c or fewer faulty units,
2. any resulting set of test outcomes, and
3. any initial assignment of enable/disable to units,

the continual application of the d-disabling rule to each unit u permanently disables u if u is faulty and permanently enables u if u is fault-free.

Fig. 1 shows a system with six units, two of which are faulty. Assume that both produce fail test outcomes of all tests they apply and that both are initially enabled. The fault-free units produce a fail test outcome if the unit tested is faulty and pass if it is fault-free. Consider the application of the 1-disabling rule to this system. If the rule is applied first to the fault-free units, they will be disabled regardless of their initial status. The subsequent application of the 1-disabling rule to the faulty units will leave them enabled. Successive applications of the 1-disabling rule will produce no change, leaving faulty units permanently enabled and fault-free units permanently disabled. Thus, the system is *not* 2-correctable using the 1-disabling rule. However, it is 1-correctable, because the first application of the 1-disabling rule to a fault-free unit u testing the single faulty unit will enable u (it fails no tests). Then, a subsequent application of the 1-disabling rule to the faulty unit disables it. Once the faulty unit is disabled, no fault-free unit is disabled. Thus, in the steady-state, all fault-free units are enabled, while the faulty unit is disabled.

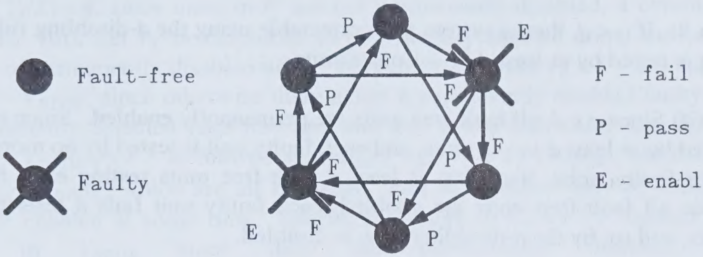


Figure 1. Application of the 1-Disabling Rule.

There is no value of d for which the system is 2-correctable using the d-disabling rule (as can be demonstrated by an exhaustive enumeration of all possibilities). However, it is 2-diagnosable [9], and so all faulty units can be uniquely identified by a central diagnoser provided there are 2 or fewer of them. Thus, distributed diagnosis places a greater restriction on the number of faulty processors which can be tolerated. It is the penalty incurred for using only local information to identify the faulty/fault-free status of units.

III. GENERAL c-CORRECTABLE SYSTEMS

We begin by showing properties possessed by every c-correctable system using the d-disabling rule.

Definition: $\Gamma(u) = \{u_i \mid u \neq u_i \in V \text{ and } u_i \text{ tests } u\}$.

$\Gamma(u)$ is the set of units that test u.

Lemma 1: Every unit in a c-correctable system using the d-disabling rule is tested by at least $d+c-1$ units.

Proof: On the contrary, suppose there exists a unit u in a c-correctable system that is tested by $d+c-2$ or fewer other units. Consider a subset $C \subseteq \Gamma(u)$ such that $|C| = c-1$. Let $F = C \cup \{u\}$ be the faulty units in the system. Assume all test results of u by units in C are pass. Then, the largest number of fault-free units testing u is $d-1$ and u, having failed less than d tests, is permanently enabled.

Thus, the system is not c-correctable.

Q.E.D.

The condition of Lemma 1 becomes necessary and sufficient when c is strictly less than d, as shown in the next lemma.

Lemma 2: If $c < d$, then a system is c-correctable using the d-disabling rule iff every unit is tested by at least $d + c - 1$ other units.

Proof: (if) Since $c < d$, all fault-free units are permanantly enabled. Since each unit is tested by at least $d + c - 1$ units, and each faulty unit is tested by no more than $c - 1$ other faulty units, there are at least d fault-free units testing each faulty unit. Since all fault-free units are enabled, each faulty unit fails d tests by enabled units, and so, by the d-disabling rule, is disabled.

(only if) On the contrary, assume there is a system that is c-correctable, but does not satisfy the condition. However, this is impossible since, by Lemma 1, all units in a c-correctable system using the d-disabling rule are tested by at least $d + c - 1$ units.

Q.E.D.

A limit on the number of units is given by:

Lemma 3: In a c-correctable system, $n \geq 2c + 1$, where n is the total number of units.

Proof: Since every faulty unit in a c-correctable system must be unambiguously identified as faulty, a c-correctable system is also c-diagnosable. From [9], a c-diagnosable system has the property $n \geq 2c + 1$.

Q.E.D.

We now show a sufficient condition for c-correctability using the d-disabling rule.

Definition: $\Gamma_{\geq d}^{-1}(Z) = \{u_i \mid u_i \notin Z \subseteq V$ and there are at least d units in Z that test $u_i\}$.

$\Gamma_{\geq d}^{-1}(Z)$ is the set of units outside of Z tested by at least d units in Z.

Theorem 1: S is c-correctable using the d-disabling rule if for all $F \subseteq V$ with $|F| \leq c$, all subsets F' of F have the property, $F' \cap \Gamma_{\geq d}^{-1}(Z) \neq \phi$, where $Z = V - F - \Gamma_{\geq d}^{-1}(F')$.

Proof: Suppose the condition holds, but S is not c-correctable. Then, either (i) there is a set of faulty units $F \subseteq V$, where $|F| \leq c$, such that there is a nonempty subset $F' \subseteq F$ consisting entirely of permanently enabled units, (ii) there is a nonempty subset $G \subseteq V - F$ of fault-free units all of which are permanently disabled, or (iii) there is a set of units $V_{NPDE} \subseteq V$ which are neither permanently disabled nor permanently enabled for some arbitrarily long sequence of applications of the d-disabling rule. In the case of (i), it must be that no unit in F' is tested by d or more permanently enabled fault-free units; that is, units in $Z = V - F - \Gamma^{-1}_{\geq d}(F')$. It follows that $F' \cap \Gamma^{-1}_{\geq d}(Z) = \phi$, a contradiction. In the case of (ii), it must be that $G \subseteq \Gamma^{-1}_{\geq d}(F')$, where F' is a set of enabled faulty units. If $Z = V - F - \Gamma^{-1}_{\geq d}(F')$, then $F' \cap \Gamma^{-1}_{\geq d}(Z) = \phi$, since units in F' are not permanently disabled, a contradiction. Consider (iii). Let V_E be the set of all permanently enabled units, and let V_D be the set of permanently disabled units. We can assume that $V_E = V - F - V_{NPDE}$ and $V_D = F - V_{NPDE}$, since otherwise there exists a permanently enabled faulty unit or a permanently disabled fault-free unit and this would fall under case (i) or (ii). Let $F' = V_{NPDE} \cap F$. It follows that $\Gamma^{-1}_{\geq d}(F') \supseteq (V - F) \cap V_{NPDE}$, since units in $(V - F) \cap V_{NPDE}$, which are all fault-free, can be disabled only by faulty units that are enabled at some time. Specifically, units in $V_D = F - F'$ cannot disable units in V_{NPDE} since they are permanently disabled. Thus, $Z = V - F - \Gamma^{-1}_{\geq d}(F') \subseteq V - F - V_{NPDE}$ consists of permanently enabled fault-free units exclusively. Since units in F' are not permanently disabled, $F' \cap \Gamma^{-1}_{\geq d}(Z) = \phi$, a contradiction.

Q.E.D.

IV. SPECIFIC c-CORRECTABLE SYSTEMS

A. COMPLETE DIGRAPHS

A *complete digraph* G(V,E) is a digraph with node set V and edge set E such that for every ordered pair (u,v) where $u,\ v \in V$, $(u,v) \in E$. We have,

Lemma 4: A complete digraph on n units is c-correctable using the d-disabling rule iff

$$c < d \leq n - c. \tag{1}$$

Proof: (only if) Let S be a complete digraph that is c-correctable using the d-disabling rule, and, assume, on the contrary, that either $c \geq d$ or $d > n - c$. Suppose $c \geq d$. Let there be c faulty units that are initially enabled, and assume that each fails all fault-free units it tests and passes all faulty units. An application of the d-disabling rule to all fault-free units will cause them to be disabled. Since there are no enabled units which fail the faulty units, an application of the d-disabling rule to faulty units leaves them enabled. This situation is permanent. Thus, the system is not c-correctable. Now suppose $d > n - c$. Consider a faulty unit u. If there are c faulty units, u is tested by $n - c$ fault-free units. If all faulty units pass all faulty units, then u is permanently enabled because there are insufficiently many fail test outcomes. Thus, S is not c-correctable using the d-disabling rule.

(if) Let $c < d \leq n-c$, and assume the system is not c-correctable using the d-disabling rule. Either there is (i) a fault-free unit that is permanently disabled, (ii) a faulty unit that is permanently enabled, or (iii) a unit that is neither permanently disabled nor permanently enabled for some arbitrarily long sequence of applications of the d-disabling rule. For a fault-free unit to be permanently disabled as in (i), it must be tested by d or more enabled faulty units. But from $c < d$, this is impossible. Thus, all fault-free units are permanently enabled. For a faulty unit to be permanently enabled as in (ii), it must be tested by no more than $d-1$ enabled fault-free units. However, this is impossible since there are at least $n-c$ permanently enabled fault-free units, and from $d \leq n-c$, it follows that each faulty unit is tested by at least d enabled fault-free units. Let V_{ED} be a nonempty set of units which are neither permanently enabled nor permanently disabled. Let V_E be the set of all permanently enabled units and V_D the set of all permanently disabled units. We can assume that $V_E \subseteq V-F$ and $V_D \subseteq F$, since otherwise this would fall under case (i) or (ii). It follows that $|V_E| \leq d-1$; otherwise all faulty units are disabled, and thus all fault-free units are enabled, which implies $V_{NPDE} = \phi$. We now show that V_{NPDE} contains no fault-free unit. On the contrary, such a unit must fail at least d tests by faulty units, which implies $d \leq c$, contradicting the condition $c < d$. Thus, V_{NPDE} contains only faulty units, and $|V_{NPDE} \cup V_D| \leq c$. However, $n = |V_{NPDE} \cup V_D| + |V_E| \leq c + d - 1$, contradicting the condition $d \leq n-c$.

Q.E.D.

B. SEGMENTED SYSTEMS

Definition: $G_{s,m}(V,E)$ is a *segmented* system if

1. $V = A_0 \cup A_1 \cup \cdots \cup A_{s-1}$,
2. $|A_i| = m$, $0 \leq i \leq s-1$,
3. $A_i \cap A_j = \phi$, $i \neq j$, $0 \leq i,j \leq s-1$, and
4. $E = \{(u,v) \mid u \in A_i$ and $v \in A_{i+1}$, where index addition is modulo $s\}$.

A segmented system consists of $s-1$ groups of m units each. The only tests that exist are between adjacent groups, in which case, all possible tests exist.

Lemma 5: A segmented system $G_{s,m}(V,E)$ is c-correctable using the d-disabling rule iff

$$\frac{c\,(2 - s \bmod 2)}{s} < d \leq m - c + 1. \qquad (2)$$

Proof: (if) On the contrary, suppose there is a segmented system where the condition holds yet is not c-correctable using the d-disabling rule. Then, either there is (i) at least one permanently enabled faulty unit, (ii) at least one permanently disabled fault-free unit, or (iii) at least one unit that is neither permanently enabled nor permanently disabled. Assume (i) holds. Let $u \in A_i$ be a permanently enabled faulty unit. Thus, there exists an integer a such that after a applications of the d-disabling rule u is always enabled. For this steady-state condition, we observe the following. *There can be at most $d - 1$ enabled fault-free units in A_{i-1}. Either there are no disabled fault-free units in A_{i-1} or there is at least one. **If there are none, A_{i-1} has at least $m - d + 1$ faulty units, and among A_{i-1} and A_i there are at least $m - d + 2$ faulty units. Thus, $m-d+2 \leq c$. However, this contradicts the rightmost inequality of (2). If A_{i-1} has at least one disabled fault-free unit, there are at least d enabled faulty units in A_{i-2}. Since A_{i-2} has at least one enabled faulty unit, a similar argument yields a contradiction to the rightmost inequality of (2) or the conclusion that A_{i-4} has d faulty units, etc.. If s is even, every other A_j can have at least d faulty units, for a total of at least $d\,s\,/\,2$ faulty units. If s is odd, every A_j can have at least d faulty units, for a total of at least $d\,s$ faulty units. Thus, $\dfrac{d\,s}{2-s \; mod \; 2} > c$, contradicting the leftmost inequality on (2). The proof for the case of (ii) is included in the above (beginning at **). Now consider case (iii). Let V_E be the set of permanently enabled units and V_D the set of permanently disabled units. We can assume $V_E = V - F - V_{NPDE}$ and $V_D = F - V_{NPDE}$, where V_{NPDE} is the nonempty set of units which are neither permanently disabled nor disabled; otherwise we have case (i) or (ii). Further, $V_{NPDE} \cap F \neq \phi$; otherwise it follows that there are no fault-free units in V_{NPDE}, since such units can be disabled only by faulty units in V_{NPDE}, which implies that $V_{NPDE} = \phi$. Let $u \in V_{NPDE} \cap F$, where $u \in A_i$. The proof that this leads to a contradiction is included in the above (beginning at *).

(only if) Assume there is a c-correctable segmented system using the d-disabling rule in which the condition does not hold. Thus, either (i) $d > m - c + 1$ or (ii) $c(2-s \mathrm{mod}\, 2)\,/\,s \geq d$. Suppose (i) holds. Let A_{i-1} contain at least $\min(m,\ c-1)$ faulty units, and let A_i contain at least one faulty unit. It follows that there are at most $m - c + 1$ fault-free units in A_{i-1}. From (i), there are fewer than d fault-free units in A_{i-1}, allowing the faulty unit in A_i to be permanently enabled. This contradicts the assumption that the system is c-correctable. Suppose (ii) holds. If s is even, there can be at least $d\,s\,/\,2$ faulty units in the system or at least d for every other A_j. Let all faulty units be initially enabled, and let each fail all tested fault-free units. Then, all units in the A_i's consisting of fault-free units exclusively are disabled, and there are no enabled fault-free units to disable the successor faulty units. Thus, all faulty units are permanently enabled, contradicting the assumption that the system is c-correctable. If s is odd, there are at least $d\,s$ faulty units in the system or at least d faulty units for every A_j. In a similar manner, it follows that all fault-free units can be permanently disabled, contradicting the assumption that the system is c-correctable.

Q.E.D.

V. CONCLUDING REMARKS

We propose a new process of self-diagnosis where the disabling of faulty units is an integral part of the diagnosis. We show conditions under which correct diagnosis is achieved; i.e. fault-free units are enabled and faulty units are disabled. The disabling mechanism, which must be done ultrareliably, is simple, so that it is constructed at reasonable cost. The approach is practical and narrows the gap between the *theory* of systems diagnosis and the practical application of that theory.

REFERENCES

[1] S. L. Hakimi and A. T. Amin, "Characterization of the connection assignment problem of diagnosable systems," *IEEE Trans. on Comput.*, vol. C-23, Jan. 1974, pp. 86-88.

[2] C. S. Holt and J. E. Smith, "Self-diagnosis in distributed systems," *IEEE Trans. on Comp.*, vol. C-34, pp. 19-32, Jan. 1985, pp. 19-32.

[3] S. H. Hosseini, J. G. Kuhl, and S. Reddy, "On self-fault diagnosis of the distributed systems," *IEEE Trans. on Comp.*, vol. C-7, Feb. 1988, pp. 248-251.

[4] S. E. Kreutzer and S. L. Hakimi, "Distributed diagnosis and the system user," *IEEE Trans. on Comp.*, vol. C-37, Jan. 1988, pp. 71-79.

[5] J. Kuhl and S. Reddy, "Distributed fault tolerance for large multiprocessor systems," *Proc. of the 7th Inter. Symp. on Comp. Architecture*, June 1980, pp. 23-30.

[6] J. Kuhl and S. Reddy, "Fault diagnosis in fully distributed systems," in *Proc. of the 11th Inter. Conf. on Fault Tolerant Computing*, June 1981, 100-105.

[7] G. Meyer and G. M. Masson, "An efficient fault diagnosis algorithm for symmetric multiple processor architectures," *IEEE Trans. on Comput.*, vol. C-27, Nov. 1978, pp. 1059-1063.

[8] R. Nair, "Diagnosis, self-diagnosis, and roving diagnosis in distributed digital systems," Coord. Sci. Lab., Univ. of Illinois, Urbana, IL Report R-823, Sept. 1978.

[9] F. Preparata, G. Metze, and R. Chien, "On the connection assignment problem of diagnosable system," *IEEE Trans. Electron. Comput.*, vol. EC-16, Dec. 1967, pp. 848-854.

[10] J. Russell and C. Kime, "System fault diagnosis: closure and diagnosability with repair," *IEEE Trans. on Comput.*, vol C-24, Dec. 1975, pp. 1078-1088.

AVERAGING THEORY FOR SIGN-SIGN LMS

B. D. O. Anderson[1], I. M. Y. Mareels[2], W. A. Sethares[4] and C. R. Johnson[2]

ABSTRACT

This paper examines the behavior of the "sign-sign" algorithm, in which the sign (or signum) function is applied to both the regressor vector and the error signal. Despite the discontinuity of the sign function, we show that averaging theory can be applied when the algorithm is driven by ergodic random inputs. The analysis defines an averaged equation around which the trajectories of the sign-sign algorithm cluster. The averaged equation is shown to converge when the algorithm is excited by Gaussian inputs. Simulations show that other input distributions cause the averaged equation to diverge.

1. INTRODUCTION

The parameter of the Least Mean Square (LMS) algorithm are updated according to [1]

$$\hat{\theta}_{k+1} = \hat{\theta}_k - \mu X_k e_k \tag{1.1}$$

where μ is a small scalar gain, X_k is the regressor vector containing time shifted versions of the inputs, and e_k is the measured error. Implementation of (1.1) imposes certain hardware demands, including a need for multipliers. To lessen such demands, several variants of (1.1) have been proposed, which replace the update kernel $X_k e_k$ by $\mathrm{sgn}(X_k)e_k$, $X_k \mathrm{sgn}(e_k)$ or $\mathrm{sgn}(X_k)\mathrm{sgn}(e_k)$. These are referred to as the signed regressor algorithm, the signed error algorithm and the sign-sign algorithms, respectively. If θ^* represents the true parameterization, then the evolution of the parameter error $\theta_k = \theta^* - \hat{\theta}_k$ for the sign-sign algorithm is defined by

$$\theta_{k+1} = \theta_k - \mu\, \mathrm{sgn}(X_k)\mathrm{sgn}(X_k'\theta_k) . \tag{1.2}$$

With stochastic inputs, LMS is well studied, the signed regressor has recently been analyzed [4] and algorithms of the form of the signed error are amenable to analysis as in [10]. This paper therefore focuses on the sign-sign algorithm and shows by example that nontrivial stochastic inputs can cause instability. In view of this possibility of divergence (for

---

[1] Department of Systems Engineering, Research School of Physical Sciences, P.O. Box 4, Australian National University, Canberra, A.C.T. 2601, Australia.

[2] School of Electrical Engineering, Phillips Hall, Cornell University, Ithaca, NY, 14853-5401, U.S.A. Supported by NSF grant, ECS-85-06149 and INT-8513400.

[3] Department of Electrical and Computer Engineering, University of Newcastle, New South Wales, Australia.

[4] Department of Electrical and Computer Engineering, University of Wisconsin, Madison, WI.

excitation sequences which pose no problem for the LMS algorithm) it becomes important to classify situations leading to acceptable behavior and to unacceptable behavior. This paper contributes to this classification by proving the stability of the sign-sign algorithm when excited by a correlated Gaussian process. In general, the application of averaging theory to an equation like

$$\theta_{k+1} = \theta_k - \mu f(X_k, \theta_k) \tag{1.3}$$

requires f (.,.) to satisfy a Lipschitz continuity condition in its second argument, see e.g. [7]. In case the X_k sequence is independent, an averaging result for f(.,.) discontinuous in θ_k is available in [9], while [10] offers an averaging result for dependent X_k and discontinuous f(.,.). This result takes some work to obtain, and, reasonably, requires side conditions (stationarity) on the X_k sequence. The result is applied in [10] to the signed error algorithm.

The method of [10] leads to results governing the probability distribution of the supremum over a finite time interval of the difference between the solution of the unaveraged equation and an averaged differential equation. Here we present a result for the supremum of the expectation of this difference, again over a finite time interval. This is of course a similar result; it is obtained with much simpler machinery. In the process, the meaning of small μ is clarified. Conventional averaging requires that μ be small. Here it only makes sense to say μ is small with respect to the magnitude of $\|\theta_k\|$. For if $\{\theta_k\}$ is a solution trajectory of the sign-sign algorithm for some μ, then $\{\lambda\theta_k\}$ is a solution trajectory when μ is replaced by $\lambda\mu$, for any positive λ, and the X_k sequence is left unaltered. The real point is that one can average and secure insights while μ is small relative to $\|\theta_k\|$. Averaging theory is thus inapplicable if θ_k tends to the vicinity of the origin. On the other hand $\|\theta_k\|$ small means that the estimate $\hat{\theta}_k$ is close to the desired parameterization θ^*.

It is worth noting that the LMS algorithm, the signed regressor, signed error and the sign-sign algorithms do not have identical convergence properties. For example, the sign-sign algorithm takes steps of magnitude $\pm\mu$, so convergence to zero is generically impossible. More importantly though, there exist periodic deterministic input sequences for which the LMS algorithm converges (with suitably small μ), and which led to divergence of the signed regressor and sign-sign algorithms, see [6] and [4]. There are also deterministic inputs for which LMS and sign-sign algorithms converge but signed-regressor diverges, and others for which LMS and signed-regressor converge but sign-sign diverges.

Section 2 defines an averaged function $f_{av}(\theta)$ (analogous to f(.,.) of (1.3) and an averaged error system with error θ^a, and then shows (under appropriate hypothesis) that the error system for the sign-sign algorithm is stable whenever the averaged error system is stable. The result is in the form of a bound on the expected difference between θ_k and θ^a.

Section 3 demonstrates that θ^a is uniformly stable when the input is a Gaussian process. Combined with the result of section 2, this shows that the estimates $\hat{\theta}_k$ of the sign-sign algorithm are convergent to a μ-region about θ^*.

Section 4 presents two sets of simulations. The first simply supports the preceding analysis by example (i.e., that $\hat{\theta}^a$ stable implies θ_k). The second set shows that θ^a is not always well behaved. By choosing certain non-Gaussian random inputs, we shown that θ^a and the sign-sign algorithm diverge together. This shows that the relationship between θ^a and the sign-sign algorithm is tighter than we were able to prove since we were unable to prove that unstable θ^a implies unstable θ. It also shows that the stability of the algorithm is highly dependent on the properties of the excitation sequence.

2. STOCHASTIC AVERAGING FOR THE SIGN-SIGN ALGORITHM

Suppose that X_k is a n-dimensional random process $X_k = (x_k, x_{k-1}, \dots x_{k-n-1})$, with x_k ergodic. Define the averaged update term as

$$f_{av}(\theta) = E\{\text{sgn } (X_k) \text{ sgn } (X_k'\theta)\} . \tag{2.1}$$

This section relates the behavior of the sign-sign error system (1.2) to the behavior of the averaged equation

$$\theta^a_{k+1} = \theta^a_k - \mu f_{av}(\theta^a_k) \tag{2.2}$$

with the additional assumption that (1.2) and (2.2) are initialized identically, that is,

$$\theta_o = \theta^a_o . \tag{2.3}$$

The initial value θ_o, if regarded as a random variable, is independent of the X_k sequence.

For n > 1 we will make several assumptions concerning f_{av} and the properties of the input process X_k. The first requires Lipschitz continuity of the averaged update $f_{av}(\theta)$.

Assumption 1: Assume that $f_{av}(\theta)$ exists for all $\theta \in \mathbf{R}^n$ and that f_{av} is Lipschitz continuous in D, an open subset of $\mathbf{R}^n/\{0\}$.

Notice that it is impossible for f_{av} to be Lipschitz continuous at the origin unless f_{av} is identically zero. An important special case is provided by a Gaussian, zero mean sequence $\{x_k\}$. Let

$$E[X_k X_k'] = R = (r_{ij}) , \tag{2.5}$$

let r_i' be the rows of R, and suppose R is positive definite. Then for a fixed but arbitrary $\bar{\theta}$, the i^{th} component of the vector $f_{av}(\bar{\theta})$ can be written (see for example [11, p. 484])

$$E(\text{sgn } X_k \text{ sgn } X_k'\bar{\theta})_i = \frac{2}{\pi} \sin^{-1}\left[\frac{r_i' \bar{\theta}}{(r_{ii})^{1/2} (\bar{\theta}' R \bar{\theta})^{1/2}}\right] = [f_{av}(\bar{\theta})]_i \tag{2.6}$$

One can verify that $f_{av}(\theta)$ is Lipschitz continuous in the set

$$D = \{\theta \in \mathbf{R}^n, \ \|\theta\| \geq \varepsilon\}$$

where ε is an arbitrary, positive constant, with a global Lipschitz constant inversely proportional to ε: notice that

$$[f_{av}(\theta)]_i = \frac{2}{\pi}\sin^{-1}\frac{(r_i'R^{-1/2})(R^{1/2}\theta)}{(\theta'R\theta)^{1/2}r_{ii}^{1/2}}$$

$$= \frac{2}{\pi}\left[\frac{\pi}{2} - \cos^{-1}\frac{(r_i'R^{-1/2})(R^{1/2}\theta)}{(\theta'R\theta)^{1/2}r_{ii}^{1/2}}\right]$$

$$= 1 - \frac{2}{\pi}\begin{pmatrix}\text{angle between vectors}\\ R^{-1/2}\ r_i \text{ and } R^{1/2}\theta\end{pmatrix} \tag{2.7}$$

In comparing (1.2) and (2.2), we shall also make assumptions on the character of the process X_k. The first concerns the direction in which the vector X_k can point, essentially, requiring that no direction is infinitely likely, in the sense that the probability density associated with the X_k direction contains no delta function.

Assumption 2: Let dS be an elemental area on the unit sphere. Then there exists $\beta > 0$ (independent of the location of dS) such that

$$\Pr\{X_k\|X_k\|^{-1}\ \varepsilon dS\} \leq \beta dS \quad \text{for every}\quad k \tag{2.8}$$

Remark: Assumption 2 is a technical assumption which insures that θ is not orthogonal to X_k infinitely often, and so prevents a "lock up" of the algorithm in some particular direction in parameter space. The only practically important stochastic processes which violate this assumption, at least in a communication or control engineering context, are those whose range consists of a finite number of points.

Another assumption on the X_k is a mixing condition. This essentially requires that information about the process at time k becomes less significant in determining the value at time $k+m$ as m becomes larger. Mixing is a way of exploiting analytically the idea that the auto correlation of a dependent process decreases with time.

Assumption 3: The following mixing properties hold:

$$\lim_{K\to\infty} E\left\{\left[\left\| \frac{1}{K}\sum_{k=mK}^{(m+1)K-1} \operatorname{sgn}X_k[\operatorname{sgn}(X_k'\theta_k)-\operatorname{sgn}(X_k'\theta_{mK})]\right\|\right] \middle| \theta_{mK} = \theta^+\right\}$$

$$= \lim_{K\to\infty} E\left\{\left[\left\| \frac{1}{K}\sum_{k=0}^{K-1} \operatorname{sgn}X_k[X_k'\theta_k)-\operatorname{sgn}(X_k'\theta_0)\right\|\right] \middle| \theta_0 = \theta^+\right\} \tag{2.10}$$

We note that we shall also be using the following particular ergodic property:

Assumption 4: In the set D, there holds

$$\lim_{K\to\infty} E\|\left[\frac{1}{K}\sum_{k=mK}^{(m+1)K-1} \operatorname{sgn}X_k \operatorname{sgn}(X_k'\theta^a_{mK}) - f_{av}(\theta^a_{mK})\right]\| = 0 \qquad (2.11)$$

In the following result, we shall establish conditions under which $E[\|\theta_k - \theta^a_k\|]$ remains small. One of these conditions is that $\|\theta^a_k\|$ does not get too small, which should be no surprise. The fact that the Lipschitz constant associated with $f_{av}(\bar{\theta})$ becomes infinite as one seeks to include arbitrarily small $\|\theta\|$ in the domain of definition suggests that one might expect difficulties in obtaining a result which permitted arbitrarily small $\|\theta^a_k\|$. Also, as mentioned in the introduction, averaging suggests that we may need μ small relative to $\|\theta^a_k\|$.

The first connection between (1.2) and (2.2) is as follows.

Proposition 1: Consider equations (1.2) and (2.2) with the same initial condition as per (2.3), and with Assumptions 1 through 4 in force. Let $\varepsilon > 0$ be arbitrary. With K sufficiently large, with μK sufficiently small and with r such that for all $m \leq r$ there holds

$$\|\theta^a_{mK}\| \geq \sigma > 2\mu K\sqrt{n} \text{ for some } \sigma$$

and for all $k \leq rK$ there holds $\theta^a_k \in D$, then

$$\sup_{0\leq k\leq rK} E\|\theta_k - \theta^a_k\| \leq \mu K \varepsilon M \qquad (2.12)$$

with

$$M = 3r\{1 + (1 + \mu KL_1)[(1 + \mu KL_1)^r - 1]\} \qquad (2.13)$$

and L independent of μ, K, r.

Remark: The minimum value which K can take depends on both the ergodic and mixing properties of $\{X_k\}$ and on ε, with smaller ε requiring a large value of K. The maximum value which μK can take depends on ε, with smaller ε requiring a smaller maximum value. It is most convenient to think of ε being chosen first, then (large enough) K, then (small enough) μ. Then r is chosen in effect so that over $[o, rK]$, the solution of the averaged equation stays in D and does not get too small.

Proof: See [16]

Proposition 1 demonstrates conditions under which the sign-sign error system (1.2) and the averaged equation (2.2) behave alike, assuming fixed μ. Note also that the solution of (2.2) can be related to the solution of a differential equation:

$$\dot{\theta}^d = -f_{av}(\theta^d) \qquad \theta^d(0) = \theta^d_o \qquad (2.22)$$

so that as $\mu \to 0$, with μK constant, $\theta^a{}_{mK} \to \theta^d(\mu mK)$. This limiting version shows that as $\mu \to 0$, $\theta_k \to \theta^a{}_k$ in a certain sense over an interval proportional to $\varepsilon\mu^{-1}$. Assume that if ε is chosen first, and K is chosen to exceed its allowed lower bound, then μ can be chosen arbitrarily small. Consider in fact ε, K fixed, and μ, r variable, with μrK fixed so that $\|\theta^a_{\mu rK}\| \simeq \|\theta^d(\mu rK)\| \geq \sigma$, and indeed $\|\theta^d(t)\| \geq \sigma$ for all $t \leq \mu rK$ and such that $\theta^a_k \in D$, $K \leq r\bar{K}$. Then as $\mu \to 0$, it is easily checked that

$$\mu K \varepsilon M \to \varepsilon(\mu Kr)\, e^{\mu Kr} . \tag{2.23}$$

Since ε is arbitrary, this shows that

$$\lim_{\mu\to 0} \sup_{0\leq k\leq rK} E\|\theta_k - \theta^a_k\| = 0$$

when μrK remains constant. We thus have

Proposition 2: Consider equations (1.2) and (2.2) with the same initial condition as per (2.3), and with Assumptions 1 through 4 in force. Suppose that the solution of (2.22) obeys $\|\theta^d(t)\| \geq \sigma > 0$ and remains in D for all $t \in [0,T]$. Then,

$$\lim_{\mu\to 0} \sup_{0 \leq k \leq \frac{T}{\mu}} E\|\theta_k - \theta^a_k\| = 0 .$$

Proposition 2 is essentially a consistency result, showing that the sign-sign algorithm and the averaged equation behave the same way for vanishing stepsizes. The result was easy to derive once the fixed stepsize case was shown in proposition 1.

3. BEHAVIOR OF THE AVERAGED EQUATION WITH GAUSSIAN INPUTS

In the previous section, we described how the solutions of the stochastic equation and the averaged equation are related. This section shows that the averaged equation converges to a region about the origin when the input is a Gaussian process. The averaged equation from (2.2) and (2.6) is

$$\theta^a_{k+1} = \theta^a_k - \mu \frac{2}{\pi} \operatorname{vec} \left[\sin^{-1} \frac{r_i \theta^a_k}{(r_{ii})^{\frac{1}{2}}(\theta^a_k{}' R \theta^a_k)^{\frac{1}{2}}}\right] \tag{3.1}$$

Now observe that because $(2/\pi)\theta \leq \sin\theta \leq \theta$, for $\theta \in [0,\pi/2]$, we have $z \leq \sin^{-1}z \leq (\pi/2)z$ for $z \in [0,1]$, while $z \geq \sin^{-1}z \geq (\pi/2)z$ for $z \in [-1,0]$. This means $\sin^{-1}z = \nu(z)z$ where $\nu(z) \in [1,\pi/2]$ for every $z \in [-1,1]$. Accordingly, (3.1) can be replaced by

$$\theta^a_{k+1} = \theta^a_k - \mu \frac{2}{\pi} \operatorname{diag}[\nu(z_1),\ldots,\nu(z_n)] \operatorname{vec}\left[\frac{r_i{}'\theta^a_k}{(r_{ii})^{\frac{1}{2}}(\theta^a_k{}' R \theta^a_k)^{\frac{1}{2}}}\right] \tag{3.2}$$

with $\nu(z_i) \in [1, \pi/2]$, and $z_i = \dfrac{r_i{}'\theta^a_k}{(r_{ii})^{\frac{1}{2}}(\theta^a_k{}' R\theta^a_k)^{\frac{1}{2}}}$

Equation (3.2) may be rewritten as follows:

$$\theta^a_{k+1} = \theta^a_k - \mu L_k R \theta^a_k (\theta^a_k{}' R \theta^a_k)^{-\frac{1}{2}} \qquad (3.3)$$

where the matrix L_k is defined as:

$$L_k = \frac{2}{\pi}\, \mathrm{diag}(\nu(z_i))\, \mathrm{diag}((r_{ii})^{-\frac{1}{2}}) \qquad (3.4)$$

To demonstrate that the averaged equation (3.1) is uniformly stable, consider the following candidate Lyapunov function:

$$V(\theta) = \theta' R \theta \qquad (3.5)$$

along the trajectories of (3.3) the difference

$$\begin{aligned} V(\theta^a_{k+1}) - V(\theta^a_k) & \\ &= -\mu\, \theta^a_k{}'(RL_kR + RL_kR)\theta^a_k(\theta^a_k{}'R\theta^a_k)^{-\frac{1}{2}} \\ &+ \mu^2\, \theta^a_k{}'RL_k\, RL_kR\theta^a_k(\theta^a_k{}'R\theta^a_k)^{-1} \end{aligned}$$

using the definition of L_k in (3.4) and the property that $\nu(z_i) \in [1, \pi/2]$, can be over bounded as

$$V(\theta^a_{k+1}) - V(\theta^a_k) \le -\mu\alpha_1 V(\theta^a_k)^{\frac{1}{2}} + \mu^2\alpha_2 \qquad (3.6)$$

where α_1 and α_2 are given by:

$$\alpha_1 = \frac{2}{\pi} \min_i [r_{ii}^{-\frac{1}{2}}]\lambda_{\min}(R)$$

$$\alpha_2 = \{\max_i [r_{ii}^{-\frac{1}{2}}]\lambda_{\max}(R)\}^2$$

This demonstrates that

$$V(\theta^a_{k+1}) - V(\theta^a_k) < 0 \quad \text{if} \quad V(\theta^a_k) > (\mu\,\alpha_2/\alpha_1)^2$$

Thus the averaged equation (3.3) is uniformly stable and $\|\theta^a_k\|$ is asymptotically of the order of μ.

From proposition 1 it follows that the trajectories of the averaged equation are on average good approximations to the trajectories of equation (2.2) over the interval $[0,N]$ provided $\|\theta^a_k\| \ge \sigma$, $k \in [0,N]$ and $\theta^a_k \in D$. In the Gaussian case, these two conditions are identical, since one can always choose D to be $\{\theta \mid \theta \in \mathbf{R}^n,\ \|\theta\| \ge \sigma,\ \|\theta\| \le \lambda^{1/2}_{\max}(R)\lambda^{-1/2}_{\min}(R)\|\theta_0\|\}$. From proposition 1, $\sigma > 2(\mu K)\sqrt{n}$, and so the sign-sign algorithm drive by a Gaussian input sequence $\{x_k\}$ on average exhibits convergent behavior for all initial conditions $\|\theta_0\| \ge \mu C$, (for some constant $C > 0$ independent of μ) and asymptotically $\|\theta_k\|$ becomes of the order of μ.

4. SIMULATION RESULTS

This section presents two sets of simulations. The first set illustrates the validity of the averaging results of the previous sections and the second set shows that the sign-sign algorithm tracks the averaged equation even when the averaged equation diverges.

Figure 1 shows simulations in which the two dimensional sign-sign algorithm was excited by i.i.d. (pseudorandom) Gaussian inputs. In each run, the two dimensional sign-sign algorithm was initialized at (2,5) in parameter error space. Each converges to a region about the origin (parameter error equal to zero) in a few hundred iterations. The steady center line is an approximation to the averaged equation θ^a, which was computed at each θ by $f_{av}(\theta) - \frac{1}{K}\sum_{i=1}^{K} \operatorname{sgn} X_i \operatorname{sign}(X_i'\theta)$ where X_i is a pseudorandum input sequence. The value K = 10,000 was found empirically to give an appropriately smooth averaging window for this calculation. The simulation shows that individual runs of the sign-sign algorithm tend to cluster around the averaged equation, as expected from proportions 1 and 2. The stepsize was set at $\mu = 0.1$.

For the second set of simulations, a non-Gaussian, dependent stochastic process was chosen. The process alternates between two nonzero mean (but independent) Gaussian random variables S1 and S2 in a Markov like manner. Let S1 be normally distributed with mean 1.5 and variance 0.1, and let S2 be normal with mean - 0.1 and variance .01. If the value of the process at time k is determined by S1, then with probability α, the value of the process at time k+1 will be determined by S1, and with probability $1-\alpha$, the value will be determined by S2. Similarly for S2 with transition probabilities β and $1-\beta$. For the simulation, $\alpha = .2$ and $\beta = .8$ were chosen. This particular process was used because it fulfills the Assumptions, (a simple Markov chain would fail Assumption 2 since X_k would point infinitely often in certain directions), and because the averaged equation diverges in simulation. It is an interesting (and non-trivial) question whether the divergence of θ^a could actually be proven.

This process was used to excite a six dimensional sign-sign algorithm. The trajectories of the averaged equation (which is computed as before) are shown in figure 2. The graph is a two dimensional slice of the six dimensional parameter error space and the trajectories of θ^a are shown for seven different starting values. In each case, the trajectory moves near the origin, and then proceeds to diverge along one of the two 45° lines. Since the function θ^a fails to be Lipschitz at the origin, it is not surprising that the solution fails to be well-defined near the origin. In the present case, there are two solutions, corresponding to the two unstable directions. It should be noted that the initial convergence to the origin in figure 2 is not a generic behavior, that is, other input sequences, other plants (values of θ^*), and other 2-D slices of the 6-D parameter error space would not necessarily show convergence to the origin before diverging. What is generic, however, it that multiple solutions of θ^a at the origin can lead to diverging "arms" in parameter error space. Moreover, this example is not singular, in the sense that a continuum of values for S1, S2, α and β lead to similar divergent behavior. It is also not hard to find multi-state Markov-like processes for which θ^a diverges in simulation.

The behavior of θ^a is only a useful guide if individual runs of the algorithm behave like the averaged equation. Figure 3 shows a number of simulations all initialized at the same location (-2, -6) in parameter error space. The individual runs tend to cluster around the averaged equation until they approach the origin, where averaging fails to hold. Then, the trajectories move out along one or the other of the unstable branches, apparently forever. These simulations indicate that the assumptions made in the theory are not restrictive, in that the behavior of the averaged equation is a good representation of the actual response, in both stable and unstable situations.

5. CONCLUSION

We have established two key facts about the sign-sign algorithm. The random trajectories of the unaveraged equation cluster around those of the averaged equation, at least until the solution of the latter becomes small, (under conditions of ergodicity, mixing, small μ etc). Second, trajectories of the averaged equation excited by Gaussian inputs converge to a ball around the origin. Combining these facts, the trajectories of the unaveraged random equation with Gaussian inputs must also converge to a region in the vicinity of the origin.

In addition, we have investigated (through simulation) the behavior of the sign-sign algorithm and the related averaged equation when excited by certain non-Gaussian processes. In all cases investigated, stability/instability of the averaged equation was coupled with stability/instability of the actual algorithm. In particular, a simple class of input processes was found for which the sign-sign algorithm diverges.

The averaged equation provides a tool which may lead to a further understanding of what types of input excitation lead to stability/instability of the sign-sign algorithm, though more insights are necessary in order to unambiguously classify all suitable and unsuitable inputs.

REFERENCES

[1] B. Widrow and S. D. Stearns, Adaptive Signal Processing, Prentice Hall, Inc., Englewood Cliffs, NJ, 1985.

[2] J. L. Moschner, 'Adaptive filter with clipped input data,' Tech. Dept. 6796-1, Information Systems Laboratory, Stanford University, 1970.

[3] R. Kumar and J. B. Moore, 'Adaptive equalization via fast quantized - stable methods,' IEEE Trans. Communications, Vol. COM-29, October 1981, pp. 492-1501.

[4] W. A. Sethares, I. M. Y. Mareels, B. D. O. Anderson, C. R. Johnson, Jr., and R. R. Bitmead, 'Excitation conditions for signed regressor least mean squares adaptation,' IEEE Trans. Circuits and Systems, June 1988.

[5] R. W. Lucky, 'Techniques for adaptive equalization of digital communication systems,' Bell Syst. Tech. J., Vol. 45, February, pp. 255-266.

[6] S. Dasgupta and C. R. Johnson, Jr., 'Some comments on the behavior of sign-sign adaptive identifiers,' Syst. Control Letters, Vol. 7, April 1986, pp. 75-82.

[7] R. Z. Khas'minskii, 'On stochastic processes defined by differential equations with a small parameter,' Theory of Probability and Its Applications, Vol. XI, 1966, pp. 211-228.

[8] R. R. Bitmead, B. D. O. Anderson and T. S. Ng, 'Convergence rate determination for gradient-based adaptive estimators,' Automatica, Vol. 22, March 1986, pp. 185-191.
[9] D. P. Derevitskii and A. L. Fradkov, 'Two models for analysing the dynamics of adaptation algorithms,' Autom. and Remote Control, Vol. 35, January 1974, pp. 59-67.
[10] A. Benveniste, M. Goursat and G. Rugt, 'Analysis of stochastic approximation schemes with discontinuous and dependent forcing terms with applications,' IEEE Trans. Auto. Control, Vol. AC-25, December 1980, pp. 1042-1058.
[11] A. Papoulis, Probability, Random Variables and Stochastic Processes, McGraw-Hill Book Co., New York, 1965.
[12] C. A. Desoer and M. Vidyasagar, Feedback Systems: Input-Output Properties, Academic Press, New York, 1975.
[13] A. Gersho, 'Adaptive Filtering with Binary Reinforcement,' IEEE Trans. on Info. Thy, Vol. IT-30, March 1984, pp. 191-198.
[14] L. Ljung and T. Soderstrom, theory and Practice of Recursive Identification, MIT Press, Cambridge, MA, 1983.
[15] N. S. Jayant and P. Noll, Digital Coding of Waveforms, Prentice Hall, 1984.
[16] B. D. O. Anderson, IMY Mareels, W. A. Sethares, C. R. Johnson, Jr., 'Averaging Theory for Sign-Sign LMS,' technical report Univ. of Wisconsin.

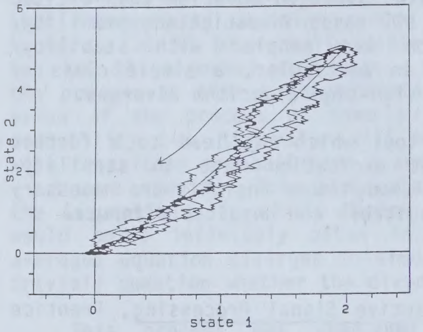


Figure 1: Trajectories and averaged trajectories of two dimensional algorithm with gaussian regressor

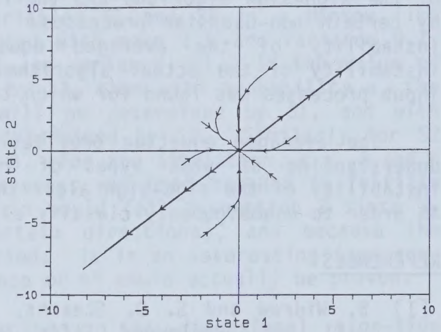


Figure 2: Two-dimensional cross section of six-dimensional algorithm exhibiting instability

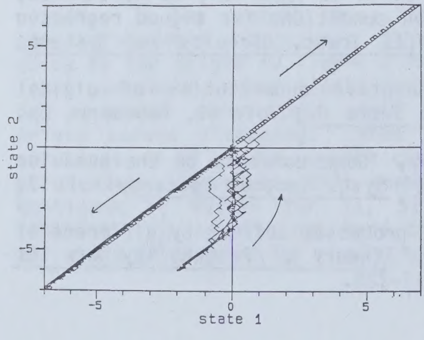


Figure 3: Unstable algorithm showing ultimate divergence

ON MEAN SQUARE CONVERGENCE OF ADAPTIVE BEAM-FORMING ALGORITHMS

G. YIN*

Department of Mathematics, Wayne State University
Detroit, MI 48202

Abstract: The asymptotic properties of adaptive beam-forming algorithms are developed in this paper. Both decreasing gains and constant gain cases are considered. By employing the methods of perturbed test functions, the mean square convergence result is obtained for the algorithm with decreasing gains, and with non-stationary and correlated signals. As for the constant gain case, a sharp bound is obtained. Moreover, the local limit behavior for the sequence of normalized errors can be investigated via the methods of weak convergence.

Keywords: adaptive beam-forming, non-stationarity, correlated signals, perturbed test functions, mean square convergence.

1. Introduction

In this work, we consider the problem of recursive adaptive beam-forming algorithms. Such problems are concerned with the determination of the azimuth of a target by utilizing a matrix composed of sensors. The outputs y_n of the individual sensors are weighted by a matrix x, such that $\{xy_n\}$ becomes the best approximation of the unknown signal $\{\psi_n\}$ in the mean square sense.

For any $n \in Z_+$, let $x \in \mathbb{R}^{r\times m}$, y_n, $C \in \mathbb{R}^{r\times l}$, and ψ_n, $\Phi \in \mathbb{R}^{m\times l}$, the basic problem is to minimize

$$E(x'y_n - \psi_n)(x'y_n - \psi_n)' \tag{1}$$

subject to constraint

$$x'C = \Phi. \tag{2}$$

It can be easily seen that the necessary and sufficient conditions for (2) to hold is that $\Phi C^\dagger C = \Phi$, where $z^\dagger$ denotes the pseudo-inverse of z.

We assume that the signals satisfy the following conditions:

(A1) Non-stationarity:

$$Ey_ny_n' = m_nQ_1 \tag{3}$$

$$Ey_n\psi_n' = m_nQ_2 \tag{4}$$

$$E\psi_n\psi_n' = m_nQ_3 \tag{5}$$

where $Q_i, i \leq 3$ are appropriate unknown constant matrices, and m_n is a sequence of positive constant.

---

* This research was supported in part by Wayne State University under the Wayne State University Research Award.

It can be shown (cf. [7] and the references therein) that there exist H, L, such that

$$Q_1 = HH', \quad Q_2 = HL', \quad Q_3 = LL'. \tag{6}$$

(A2) Correlations:

$$\sum_{n=1}^{\infty} a_n(y_n y_n' - m_n HH') \text{ converges a.s.} \tag{7}$$

$$\sum_{n=1}^{\infty} a_n(y_n \psi_n' - m_n HL') \text{ converges a.s.} \tag{8}$$

where

$$a_n > 0 \quad a_n m_n \to 0 \quad \sum_{n=1}^{\infty} a_n m_n = \infty. \tag{9}$$

The optimal x^*, which satisfies (1) and (2) is given by (cf. [2], [7])

$$\begin{aligned} x^* &= C^{\dagger'}\Phi' + (Pm_nQ_1P)^{\dagger}(m_nQ_2 - m_nQ_1C^{\dagger'}\Phi') \\ &= C^{\dagger'}\Phi' + (PHH'P)^{\dagger}(HL' - HH'C^{\dagger'}\Phi') \end{aligned} \tag{10}$$

where $P = I - CC^{\dagger}$.

Since H, L are unknown, to obtain x^* from (10) directly is impossible. Even if we can estimate H, L, it is rather time consuming and expensive to compute the pseudo-inverse sequentially, for large dimensional systems. Therefore, we shall approximate x^* by a matrix x_n at each time n, such that x_n can be corrected on the basis of the measurement y_n. The approximation procedure is given by

$$x_{n+1} = C^{\dagger'}\Phi' + P[x_n + a_n(y_n\psi_n' - y_n y_n' x_n)], \quad x_0 = C^{\dagger'}\Phi'. \tag{11}$$

(11) is called an adaptive beam-forming algorithm. When the constraint vanishes, it reduces to the usual adaptive filter algorithm. As a result, the adaptive beam-forming algorithm can be called an adaptive filter algorithm with constraints.

Due to wide range of applicability in regulation and tracking, adaptive control, optimization, under water signal processing and in many other related fields, adaptive filter algorithms and their constrained counter part adaptive beam-forming algorithms have been studied extensively, for example [1]-[14] among others.

In the early stage, the signals were assumed to be independent. In most applications, this kind of condition is far from being fulfilled. The signals we are dealing with are normally neither independent nor uncorrelated. To search for more realistic conditions thus becomes an important task. Ljung [4], Kushner and Clark [5] used the so called "ODE approach" to deal with problem of this sort. They managed to relax the conditions on the noise, allow the noise to be correlated. However, in order to get the w.p.1 convergence result, a condition of the iterates x_n returning to a bounded set infinitely often is needed. [6] used a projection scheme, which requires a priori knowledge of the bounded projection region. The boundedness conditions were eliminated in [8-9], However, their assumptions are quite strong otherwise. They required that the moments of y_n up to the order of 24 be bounded, and the $4 + \alpha$ th moments of ψ_n be bounded. From an application point of view, these conditions are rather restrictive.

Progress has also been made for the development of constrained adaptive filter or adaptive beam-forming algorithms, for example [2], [7], [12]-[14] among others. [2], [7] laid out a good foundation for our current studies, but only independent signals were treated. [12] considered the case of nonstationary and correlated signals, but the author required the second moments to be bounded below and above. To avoid using the boundedness condition, a random truncation method was employed in [13]. Recently, we proved the strong consistency for the algorithm via a direct method. To the author's knowledge, the conditions in [14] are the weakest up to date for this kind of algorithm.

In this paper, we are shall prove the second order or mean square convergence for the adaptive beam-forming algorithm, and obtain some asymptotic results for the normalized errors.

We organize this paper as follows. In next section, we give description of the algorithm, and review the w.p.1 convergence result briefly. In section 3, we derive the mean square convergence result via the technique of perturbed test functions, for the decreasing gain case. Finally, we close this paper by making some remarks on algorithms with constant gain in section 4.

To proceed, a word about the notations is in order in the sequel, we shall always use K to denote a positive constant, its value may change from usage to usage.

2. Review of w.p.1 convergence result

Let $S_n = Px_n$. Since $S'_n = x'_n(I - CC^\dagger) = x'_n - \Phi C^\dagger$, to study the convergence of x_n, we need only consider the convergence of S_n. It follows from (11)

$$S_{n+1} = S_n + a_n P[y_n \psi'_n - y_n y'_n (S_n + C^{\dagger'} \Phi')], \quad S_0 = 0 \tag{12}$$

$$S^* = (PHH'P)^\dagger (HL' - HH'C^{\dagger'} \Phi'). \tag{13}$$

It was demonstrated in [7],

$$S_n \in L(PH) \text{ a.s. } \forall n \geq 0 \quad Py_n \in L(PH) \text{ a.s. } \forall n \geq 0 \tag{14}$$

where $L(PH)$ is the vector subspace spanned by the columns of PH. In fact, we need only consider the case $PH \neq 0$, for otherwise we have $S_n = 0$ a.s. Suppose that the columns of V form an orthonormal basis for $L(PH)$, then for any $n \geq 0$, there is a unique matrix $O_n \in \mathbb{R}^{dimL(PH) \times r}$, such that $S_n = VO_n$ a.s. Which in turn implies that

$$VO_{n+1} = VO_n + a_n P[y_n \psi'_n - y_n y'_n (VO_n + C^{\dagger'} \Phi')].$$

Notice, we also have $PVO_n = PS_n = S_n = VO_n$. Hence

$$\begin{aligned} O_{n+1} &= O_n + a_n V'P[y_n \psi_n' - y_n y_n'(VO_n + C^{\dagger'} \Phi')] \\ &= (I - a_n V'P y_n y_n' PV) O_n + a_n V'P(y_n \psi_n' - y_n y_n' C^{\dagger'} \Phi') \\ &= O_n - a_n z_n z_n' O_n + a_n G_n \end{aligned} \tag{15}$$

where

$$z_n = V'Py_n \qquad G_n = V'P(y_n \psi_n' - y_n y_n' C^{\dagger'} \Phi').$$

Consequently, we need only study the modified algorithm (15). The problem thus reduces to prove $O_n \to O^*$, with

$$O^* = V'(PHH'P)^{\dagger}(HL' - HH'C^{\dagger'} \Phi').$$

Remark: We shall comment on our assumptions (A1) and (A2) briefly. In (5)-(7), the appearance of m_n reflects that the signals y_n and ψ_n are changing synchronously, which is not restrictive. Boundedness condition need not be assumed for either the processes or their moments. (A2) covers a large class of random processes. For instance, $\{y_n y_n' - m_n HH'\}$, $\{y_n \psi_n' - m_n HL'\}$ can be ARMA models or MA process with infinitely many terms. Thus, the signals considered in [8] is actually a special case of ours. Note that m_n can decay to 0 or diverge to ∞; (9) indicates the conditions on the rate of decay or divergence.

Theorem 1. *Under conditions (A1)-(A2), the sequence given by (15) is strongly consistent. I.e.,*

$$O_n \to O^* \text{ a.s.}$$

The proof of this theorem consists of proofs of the following lemmas; the proofs can be found in [14].

Lemma 1. $\sum_{n=1}^{\infty} a_n(G_n - z_n z_n' O^*)$ *converges a.s.*

Lemma 2. *There exist a* $\eta > 0$ *and a* K_η, *such that* $\eta\lambda < 1$ *and*

$$\left| \prod_{j=n}^{N(n,\eta)} (I - a_j z_j z_j') \right| < 1 - \frac{1}{2}\eta\lambda \text{ a.s. } \forall n > K_\eta$$

where λ *is the smallest eigenvalue of* $V'PHH'PV$ *(note* $V'PHH'PV$ *is a positive definite matrix), and* $N(n,\eta) = \max\{l, \sum_{j=n}^{l} a_j m_j \le \eta\}$.

Lemma 3. *For fixed* ω, *there exist* η *and* K_η, *which satisfy the conditions of Lemma 2, such that for any* $n \ge K_\eta$ *and* $l \ge N(n,\eta)$,

$$\left| \prod_{j=n}^{l} (I - a_j z_j z_j') \right| \le e^{\frac{\lambda}{2}(\eta - \sum_{j=n}^{l} a_j m_j)}.$$

Lemma 4. *$\{O_n\}$ is bounded a.s.*

3. Mean square convergence of $O_n - O^*$

In this section, we shall show that the mean square deviation tends to 0 as n tends to infinity by using the idea of perturbed test function. The methods of perturbed test function were introduced in [15] and [16].

To start with, rewrite (14) as

$$\begin{aligned} O_{n+1} - O^* &= O_n - O^* - a_n V'Py_ny_n'PV(O_n - O^*) \\ &\quad + a_nV'P(y_n\psi_n' - y_ny_n'C^{\dagger'}\Phi' - y_ny_n'PVO^*) \\ &= O_n - O^* - a_nm_nV'PHH'PV(O_n - O^*) \\ &\quad + a_nV'P(y_ny_n' - m_nHH')PV(O_n - O^*) \\ &\quad + a_nV'P(y_n\psi_n' - y_ny_n'C^{\dagger'}\Phi' - y_ny_n'PVO^*). \end{aligned} \tag{16}$$

In order to get a mean square convergence result for the sequence $O_n - O^*$, it suffices to consider the mean square convergence for the sequence R_n, where R_n is defined by $R_n = (O_n - O^*)b$, for an arbitrary $b \in \mathbb{R}^r$. We have

$$\begin{aligned} R_{n+1} = R_n - a_nm_nV'PHH'PVR_n + a_nV'P(y_ny_n' - m_nHH')PVR_n \\ + a_nV'P(y_n\psi_n' - y_ny_n'C^{\dagger'}\Phi' - y_ny_n'PVO^*)b. \end{aligned} \tag{17}$$

To get the desired bound, we need to replace (A2) by a slightly stronger condition (A2′).

(A2′) Let E_n denote the conditional expectation with respect to the σ-algebra $\{x_0, y_j, \psi_j, j \leq n\}$. The following equations hold uniformly in n.

$$\sum_{j=n}^{\infty} a_j|E_n(y_jy_j' - m_jHH')| = O(a_nm_n) \tag{18}$$

$$\sum_{j=n}^{\infty} a_j|E_n(y_j\psi_j' - m_jHL')| = O(a_nm_n) \tag{19}$$

where

$$a_j > 0, \quad a_jm_j \xrightarrow{j} 0, \quad \sum_{j=1}^{\infty} a_jm_j = \infty \quad \sum_{j=1}^{\infty} a_j^2m_j^2 < \infty. \tag{20}$$

This condition does not seem to be restrictive. In fact, if $m_n = c$ a constant, and

$$\sum_{j=n}^{\infty} |E_ny_jy_j' - cHH'| \quad \sum_{j=n}^{\infty} |E_ny_j\psi_j' - cHL'|$$

are bounded, which are essentially one of the conditions used in [10], then (A2′) is satisfied. It can be seen that if $\{m_n\}$ is bounded, together with the boundedness of the above sums, then (A2′) is also satisfied. Moreover, (A2′) also includes the case of unbounded $\{m_n\}$. Notice that [9] also dealt with the second order convergence problem. Apart from the finite moment assumptions, they only allow stationary signals (in fact, a subset of strong mixing processes). Our hypotheses are much more general and weaker.

Theorem 2. *Under the conditions (A1)(A2′), we have*

$$E|R_n|^2 \to 0 \text{ as } n \to \infty.$$

The arbitrariness of b then implies

Corollary. *If the conditions of Theorem 2 are satisfied, then*

$$E|O_n - O^*|^2 \to 0 \text{ as } n \to \infty.$$

Remark: The result tells us that the mean square deviation of the iterates O_n from the optimal O^* is asymptotically 0, or equivalently, the deviation of x_n from x^* is asymptotically 0.

To prove Theorem 2, we define $v(R) = R'R$. By means of truncated Taylor expansion, we get

$$\begin{aligned} E_n v(R_{n+1}) - v(R_n) &= E_n R_n'(R_{n+1} - R_n) + O(a_n m_n)(1 + v(R_n)) \\ = R_n'[-a_n m_n V'PHH'PVR_n &+ a_n V'P(y_n y_n' - m_n HH')PVR_n \\ &+ a_n V'P(y_n \psi_n' - y_n y_n' C^{\dagger'} \Phi' - y_n y_n' PVO^*)b] + O(a_n^2 m_n^2)(1 + v(R_n)). \end{aligned} \tag{21}$$

We define the following two perturbations. These perturbations will serve the purpose of averaging out the "left over" terms in (21).

$$u(n, R) = \sum_{j=n}^{\infty} E_n R' a_j V'P(y_j y_j' - m_j HH')PVR$$

$$w(n, R) = \sum_{j=n}^{\infty} E_n R' a_j V'P(y_j \psi_j' - y_j y_j' C^{\dagger'} \Phi' - y_j y_j' PVO^*)b.$$

By condition (A2′), we have

$$|u(n, R)| \le K a_n m_n (1 + v(R)). \tag{22}$$

As for $w(n, R)$, we have

$$\begin{aligned} w(n, R) = \sum_{j=n}^{\infty} & E_n R' a_j m_j V'P(HL' - HH'C^{\dagger'}\Phi' - HH'PVO^*)b \\ &+ \sum_{j=n}^{\infty} E_n a_j R'V'P(y_j \psi_j' - HL')b \\ &+ \sum_{j=n}^{\infty} E_n R' a_j V'P(HH' - y_j y_j')C^{\dagger'}\Phi' b \\ &+ \sum_{j=n}^{\infty} E_n R' a_j V'P(HH' - y_j y_j')PVO^* b. \end{aligned} \tag{23}$$

The last three terms on the right hand side of (23) are of the order $O(a_n m_n)(1+v(R))$.

Since (see [7] or [12]) for any A,

$$(PHH'P)(PHH'P)^{\dagger}A = PA \text{ and } VV' = I$$

$$\begin{aligned} V'PHH'PVO^* &= V'(PHH'P)VV'(PHH'P)^{\dagger}(HL - HH'C^{\dagger'}\Phi') \\ &= V'P(HL - HH'C^{\dagger'}\Phi') \end{aligned} \tag{24}$$

the first term on the right hand side of (23) equal to 0, and hence,

$$|w(n,R)| \le Ka_n m_n(1+v(R)) \tag{25}$$

Now, define

$$W(n,R) = v(R) + u(n,R) + w(n,R). \tag{26}$$

It follows that

$$\begin{aligned} E_nW(n+1,R_{n+1}) - W(n,R_n) &= E_nv(R_{n+1}) - v(R_n) \\ &+ E_nu(n+1,R_{n+1}) - E_nu(n+1,R_n) + E_nu(n+1,R_n) - u(n,R_n) \\ &+ E_nw(n+1,R_{n+1}) - E_nw(n+1,R_n) + E_nw(n+1,R_n) - w(n,R_n). \end{aligned} \tag{27}$$

Note

$$E_nu(n+1,R_n) - u(n,R_n) = -a_nR_n'V'P(y_ny_n' - m_nHH')PVR_n \tag{28}$$

$$E_nw(n+1,R_n) - w(n,R_n) = -a_nR_n'V'P(y_n\psi_n' - y_ny_n'C^{\dagger'}\Phi' - y_ny_n'PVO^*)b. \tag{29}$$

By similar kind of argument as in (23)-(25), we can show that

$$|E_nu(n+1,R_{n+1}) - E_nu(n+1,R_n)| = O(a_n^2m_n^2)(1+v(R_n))$$

$$|E_nw(n+1,R_{n+1}) - E_nw(n+1,R_n)| = O(a_n^2m_n^2)(1+v(R_n))$$

Recall that $V'PHH'PV$ is positive definite, and λ is the smallest eigenvalue of this matrix. By virtue of the inequalities (23) and (25),

$$\begin{aligned} &EW(n+1,R_{n+1}) - EW(n,R_n) \\ &\le -\lambda a_nm_nEv(R_n) + O(a_n^2m_n^2)(1+Ev(R_n)) \\ &\le -\lambda a_nm_nEW(n,R_n) + O(a_n^2m_n^2)(1+EW(n,R_n)). \end{aligned} \tag{30}$$

Hence,

$$EW(n+1,R_{n+1}) \le (1 - \lambda a_nm_n + Ka_n^2m_n^2)EW(n,R_n) + O(a_n^2m_n^2). \tag{31}$$

Owing to the properties of a_n, m_n, we can certainly find $t > 0$, such that

$$\lambda a_n m_n - K a_n^2 m_n^2 \geq t a_n m_n$$

and hence,

$$EW(n+1, R_{n+1}) \leq (1 - t a_n m_n) EW(n, R_n) + O(a_n^2 m_n^2). \tag{32}$$

Iterate inequality (32), we get

$$\begin{aligned} EW(n+1, R_{n+1}) \leq & \prod_{j=0}^{n} (1 - t a_j m_j) EW(0, R_0) \\ & + K \sum_{k=0}^{n} \prod_{j=k+1}^{n} (1 - t a_j m_j) a_k^2 m_k^2. \end{aligned} \tag{33}$$

This together with (23) and (25) then yield for some $t_1 > 0$,

$$\begin{aligned} Ev(R_{n+1}) \leq & K \prod_{j=0}^{n} (1 - t_1 a_j m_j) Ev(R_0) + K \sum_{k=0}^{n} \prod_{j=k+1}^{n} (1 - t_1 a_j m_j) a_k^2 m_k^2 \\ \leq & K exp \left(-t_1 \sum_{j=0}^{n} a_j m_j \right) Ev(R_0) + K \sum_{k=0}^{n} exp \left(-t_1 \sum_{j=k+1}^{n} a_j m_j \right) a_k^2 m_k^2. \end{aligned} \tag{34}$$

Now, letting $n \to \infty$, the asserted result follows.

4. An Algorithm with constant gain

Problem with constant gain can also be dealt with as well. In this case, in lieu of using a_n, we will use a small gain parameter ϵ. Instead of (11), we study the following algorithm

$$x_{n+1}^{\epsilon} = C^{\dagger'} \Phi' + P\Big(x_n^{\epsilon} + \epsilon y_n^{\epsilon} (\psi_n^{\epsilon'} - y_n^{\epsilon'} x_n^{\epsilon})\Big), \; x_0 = C^{\dagger'} \Phi' \tag{35}$$

with x^* given by (10) as before.

In the process of getting the L^2 bound, we will replace the conditions (A1)(A2$'$) by

(A3) The following equations hold

$$E y_n^{\epsilon} y_n^{\epsilon'} = m^{\epsilon} Q_1 \tag{36}$$

$$E y_n^{\epsilon} \psi_n^{\epsilon'} = m^{\epsilon} Q_2 \tag{37}$$

$$E \psi_n^{\epsilon} \psi_n^{\epsilon'} = m^{\epsilon} Q_3 \tag{38}$$

where $m^\epsilon > 0$, such that $\epsilon m^\epsilon \to 0$, as $\epsilon \to 0$, and $Q_i, i \leq 3$ are appropriate unknown constant matrices, with

$$Q_1 = HH' \quad Q_2 = HL' \quad Q_3 = LL'.$$

(A4) The following equations hold uniformly in n.

$$|\sum_{j=n}^{\infty} E_n(y_j^\epsilon y_j^{\epsilon\prime} - m^\epsilon HH'| = O(m^\epsilon) \tag{39}$$

$$|\sum_{j=n}^{\infty} E_n(y_j^\epsilon \psi_j^{\epsilon\prime} - m^\epsilon HL')| = O(m^\epsilon). \tag{40}$$

Under these conditions, by performing the same kind of analysis as we did in the last section, we get a sharp bound on the second moment. Namely if n is large enough, then the deviation of the second moment is of the order ϵm^ϵ. The statement is read as follows.

Theorem 3. *If conditions (A3)(A4) are satisfied, then there is a $N_\epsilon > 0$, such that for all $n \geq N_\epsilon$, $E|O_n^\epsilon - O^*|^2 = O(\epsilon m^\epsilon)$. where O_n^ϵ, O^* can be obtained along the same line of the last section.*

Moreover, we can replace m^ϵ in the assumptions by a sequence m_n^ϵ indexed by both ϵ and n, such that for any fixed ϵ, as $n \to \infty$, $m_n^\epsilon \to m^\epsilon$, then the aforementioned result still holds. To get better feeling of the asymptotic properties, we consider the normalized error $U_n^\epsilon = \frac{O_n^\epsilon - O^*}{\sqrt{\epsilon m^\epsilon}}$. Under suitable conditions, by using methods of weak convergence, we can show that appropriate interpolated sequence of U_n^ϵ converges to a Gauss Markov process weakly. This extends the result of [10]. In fact, their result corresponds to ours with $m^\epsilon = 1$.

REFERENCES

[1] B. Widrow, P.E. Mantey, L.T. Griffiths and B.B. Goode, Adaptive antenna systems, *Proc. IEEE*, 55(1967), pp2143-2158.

[2] O.I. Frost, An algorithm for linearly constrained adaptive array processing, *Proc. IEEE*, 60(1972), pp926-935.

[3] B. Widrow, J.M. McCool, M.G. Larimore and C.R. Johnson, Stationary learning characteristics of the LMS adaptive filter, *Proc. IEEE*, 64(1975), pp1692-1716.

[4] L.Ljung, Analysis of stochastic approximation algorithm, *IEEE Transactions on Automatic Control*, AC-22 (1977), pp551-575.

[5] H.J. Kushner and D.S. Clark, *Stochastic Approximation Method for Constrained and Unconstrained Systems*, Springer-Verlag, 1978.

[6] H.J. Kushner, A projected stochastic approximation method for adaptive filters and identifiers, *IEEE Transactions on Automatic Control*, AC-25 (1980), pp836-838.

[7] H.F. Chen, A recursive algorithm for adaptive beam-formers, *Ker Xue Tongbao*, 26:6 (1981), pp490-493.

[8] E. Eweda and O. Macchi, Quadratic mean and almost sure convergence of unbounded stochastic approximation algorithm with correlated observations, *Ann. Inst. Henri Poincare*, Section B, vol. 14, No 3 (1983), pp235-255.

[9] O. Macchi and E. Eweda, Second order convergence analysis of stochastic adaptive linear filter, *IEEE Transactions on Automatic Control*, AC-28(1983), pp76-86.

[10] H.J. Kushner and A. Shwartz, Weak convergence and asymptotic properties of adaptive filters with constant gains, *IEEE Transactions on Information Theory*, IT-30(1984), pp177-182.

[11] A.Z. Di, A new result on the iterative algorithm for adaptive array processing with correlated and nonstationary noise, *Journal of Systems Sciences and Mathematical Sciences*, 5(1985), pp151-160.

[12] H.F. Chen, *Recursive Estimation and Control for Stochastic Systems*, John Wiley & Sons, 1985.

[13] Y.M. Zhu, An algorithm with randomly varying truncations for adaptive beam-formers, *Applied Mathematics and Mechanics*, 7(1986), pp435-442.

[14] Y.M. Zhu and G. Yin, Adaptive filter with constraint and correlated nonstationary signals, *Systems & Control Letters*, 10(1988), pp271-279.

[15] G.B. Blankenship and G.C. Papanicolaou, Stability and control of stochastic systems with wide band noise, *SIAM J. Appl. Math.*, 34(1978), pp437-476.

[16] H.J. Kushner, *Approximation and Weak Convergence Methods for Random Processes with Applications to Stochastic Systems Theory*, MIT Press, 1984.

AN EFFICIENT ALGORITHM FOR TRACKING MULTIPLE MOVING TARGETS*

C. R. SASTRY, E. W. KAMEN and M. SIMAAN
Signal Processing and Interpretation Laboratory
Department of Electrical Engineering
University of Pittsburgh
Pittsburgh, PA 15261

ABSTRACT

In this paper a new algorithm for tracking multiple moving targets is presented. The tracking problem is formulated with a linear array of sensors and assumes a narrow band representation for the moving target signals. Initial estimates of the number of targets, the sensor noise level and the angles-of-arrival of the targets are obtained using any of the high resolution signal subspace methods such as MUSIC. These estimates are then used to initiate the tracking procedure. Updated estimates of the angles-of-arrival, as the targets move, are obtained by minimizing a matrix norm. A closed form expression for the gradient vector of this norm with respect to the angles-of-arrival is derived. Using this expression a conjugate gradient iterative algorithm is implemented at the end of successive measurement intervals to minimize the norm. We show that if this procedure is initiated with the estimated angles-of-arrival of the previous interval, tracking can be achieved. In the special case of uncorrelated sources, the results obtained are shown to be independent of the noise power. A simulation example is presented to illustrate the results.

I. INTRODUCTION

The problem of target tracking involves estimating the trajectories of targets as they move in space. Estimates of target states (i.e. position and velocities) are generated from a set of measurements provided by one or more sensors. The problem of tracking a single target has been tackled successfully by many authors [1]. In the multitarget case a key problem is to determine which measurement in the measurement set corresponds to which target. This data association problem has been approached using a number of techniques such as track splitting algorithms [2], maximum likelihood methods [3] and probabilistic data association methods [4, 5]. In this paper, we present a different approach to the problem of tracking multiple moving targets. This approach takes advantage of the MUSIC algorithm which is a high resolution algorithm [6, 7] for estimating the angles-of-arrival of signals emitted from stationary targets.

One possible approach for tracking moving targets is to repeat the use of the MUSIC algorithm. However this would require associating successive estimates of the angles-of-arrival. Assuming that the source covariance matrix does not change, association of updated estimates can be achieved by comparing permutations of the columns in the source covariance matrix. However this would necessitate *N!* (*N*=number of targets) comparisions between two sets of estimates to establish the correct association. Thus, although the MUSIC algorithm performs very well in the classical angles-of-arrival estimation problem, it cannot be used to track moving sources unless a technique for associating estimates is considered. In contrast, in our approach the updated estimates are automatically associated with the previous estimates.

---

*This work was supported in part by the SDIO/IST and managed by the office of Naval Research under Contract No. N00014-86-K-0543.

Essentially, our approach requires the implementation of the MUSIC algorithm only once after the first set of measurements are obtained. Estimates from the MUSIC algorithm are then used to initiate an iterative minimization algorithm which provides updated estimates of the angles-of-arrival based on data collected during the next measurement interval. The minimization algorithm performs a least squares fit of the data to a matrix whose entries are functions of the angles-of-arrival. This algorithm is repeated at the end of each measurement interval with initial estimates equal to the estimates obtained during the previous measurement interval. The procedure achieves the desired objective of updating estimates of the angles-of-arrival while maintaining the correct association between updated estimates and previous estimates.

In the next section, we discuss the signal model and sensor geometry used in our work. In section III we outline the MUSIC algorithm. In section IV we present our approach and discuss some of its features. Finally, in section V we illustrate the performance of our algorithm with a simulation example.

II. SIGNAL MODEL AND SENSOR GEOMETRY

Consider a uniformly spaced linear array consisting of M sensors as shown in figure(1). Also consider N targets moving with arbitary trajectories in the x-y plane and emitting plane waves that impinge on the array from distinct directions. Note that the angles-of-arrival are functions of time, and we are interested in tracking these angles as they change with the movement of the targets. Assume that the plane waves are narrowband having the same carrier frequency centered at f_0. Following Haykin [8], we express the signal produced by the i^{th} source as:

$$s_i(t) = A_i(t) \cos\left(2\pi f_0 t + \phi_i(t)\right); \qquad i=1, 2, ..N. \tag{1}$$

where $A_i(t)$ and $\phi_i(t)$ are slowly varying functions of time, denoting the amplitude and phase of the signal $s_i(t)$ respectively.

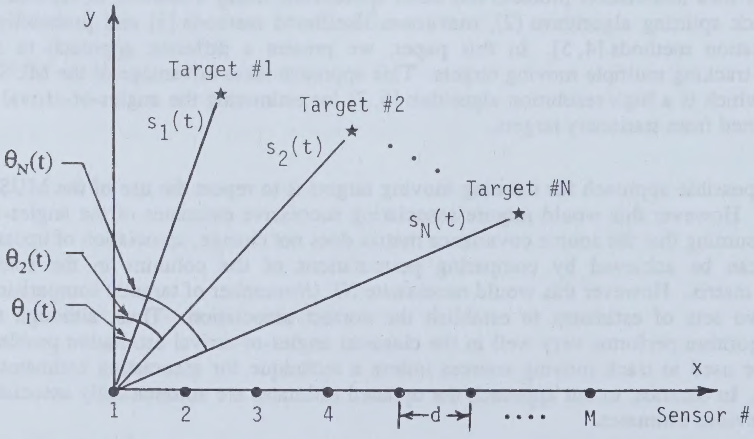


Figure 1. Sensor geometry used for tracking

With sensor#1 as reference, the time delay τ_{ik} of the i^{th} plane wave between when it reaches the first sensor and when it reaches the k^{th} sensor is given by:

$$\tau_{ik} = \frac{d}{\lambda f_0}(k-1)\, sin\theta_i(t)\,, \quad i=1, ..N, \text{ and } k=1, ...M \tag{2}$$

where d is the sensor spacing, $\lambda = c/f_0$, and c is the signal propagation velocity. Thus the signal from the i^{th} source as received by the k^{th} sensor can be modelled as:

$$s_{ik}(t) = A_i(t)\, cos\left(2\pi f_o t - 2\pi(k-1)\frac{d}{\lambda} sin\theta_i(t) + \phi_i(t)\right);i=1, ..N \text{ and } k=1, ...M. \tag{3}$$

Representing the signal and noise by their complex envelopes, the total signal received at the k^{th} sensor from all targets can be written as:

$$r_k(t) = \sum_{i=1}^{N} a_i(t)\, e^{\{-j2\pi(d/\lambda)(k-1)sin\theta_i(t)\}} + v_k(t)\ ;\ k=1, ...M \tag{4}$$

where $a_i(t)=A_i(t)e^{j\phi_i(t)}$ is the complex envelope of the signal emitted by the i^{th} target and $v_k(t)$ is a complex-valued random measurement noise at the k^{th} sensor with zero mean and variance equal to σ^2. We assume that the sensor noises are uncorrelated with the signals and among themselves.

Writing (4) in matrix form we have:

$$\mathbf{r}(t) = \mathbf{B}(t)\, \mathbf{a}(t) + \mathbf{v}(t) \tag{5}$$

where **r**(t), **a**(t) and **v**(t) are vectors defined as :

$$\mathbf{r}(t)=[r_1(t), r_2(t), ...r_M(t)]^T$$
$$\mathbf{a}(t)=[a_1(t), a_2(t), ...a_N(t)]^T$$
$$\mathbf{v}(t)=[v_1(t), v_2(t), ...v_M(t)]^T$$

and **B**(t) is an $(M \times N)$ angles-of-arrival matrix given by:

$$\mathbf{B}(t) = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ e^{-j\mu_1(t)} & e^{-j\mu_2(t)} & \cdots & e^{-j\mu_N(t)} \\ \vdots & \vdots & \ddots & \vdots \\ e^{-j(M-1)\mu_1(t)} & e^{-j(M-1)\mu_2(t)} & \cdots & e^{-j(M-1)\mu_N(t)} \end{bmatrix} \tag{6}$$

with $\mu_i(t) = 2\,\pi\,\frac{d}{\lambda}\, sin\theta_i(t)$. Note the Vandermonde structure of $\mathbf{B}(t)$. The spatial correlation matrix of the observed signal vector **r**(t) is called the sensor covariance matrix and is given by:

$$\begin{aligned} R(t) &= E\,[\mathbf{r}(t)\, \mathbf{r}^{+}(t)\,] \\ &= \mathbf{B}(t)\, R_a(t)\, \mathbf{B}^{+}(t) + \sigma^2 I \end{aligned} \tag{7}$$

Where E denotes the expectation operator and the superscript + denotes complex conjugate transpose. The matrices $R_a(t)$ and $\sigma^2 I$ are the spatial correlation matrices of the signal and noise vectors respectively and are given by:

$R_a(t) = E\,[\mathbf{a}(t)\,\mathbf{a}^{+}(t)\,]$ and $\sigma^2 I = E\,[\,\mathbf{v}(t)\,\mathbf{v}^{+}(t)\,]$.

The sensor covariance matrix is estimated using time averaging. If we assume that the angles-of-arrival of the targets do not change much within each interval $[nT, (n+1)T]$ for n=0, 1,..., then we can obtain estimates of $R(t)$ at instants $[T, 2T,...]$ as:

$$R(nT) \approx \hat{R}(nT) = \frac{1}{Q}\sum_{k=1}^{Q} \mathbf{r}\Big(nT - \frac{kT}{Q}\Big)\mathbf{r}^{+}\Big(nT - \frac{kT}{Q}\Big), \quad n=1, 2, ... \tag{8}$$

where Q is the number of discrete measurements within each interval.

III. MUSIC Algorithm

For the MUSIC algorithm we assume that the targets are stationary. The sensor covariance matrix is given by:

$$R = \mathbf{B}\,R_a\,\mathbf{B}^{+} + \sigma^2 I \tag{9}$$

and the time averaging equation (8) for estimating R is given by:

$$R \approx \hat{R} = \frac{1}{Q}\sum_{k=1}^{Q} \mathbf{r}(k)\,\mathbf{r}^{+}(k) \tag{10}$$

Note that here $\mathbf{B}$ is constant within each processing interval. The problem now is to determine the matrix $\mathbf{B}$ given the measurements, $\mathbf{r}(k)$, k=1,2,....Q. Using (9) and assuming that the columns of $\mathbf{B}$ are linearly independent the MUSIC algorithm [6] can be briefly summarized as follows:

1) Compute the sensor covariance matrix $\hat{R}$ using equation (10).

2) Determine the eigenvalues and eigenvectors of $\hat{R}$

3) Identify the minimal eigenvalue and its multiplicity and set

minimal eigenvalue ($\lambda_{\min}$) = noise power (σ^2), and
Number of sources (N) = M – (multiplicity of $\lambda_{\min}$)

4) Let $\mathbf{c}(\theta) = [1, e^{-j\mu}, ...e^{-j(M-1)\mu}]^T$. We define the function :

$$q(\theta) = \frac{1}{\mathbf{c}^{+}(\theta)\,E\,E^{+}\,\mathbf{c}(\theta)}. \tag{11}$$

where E is the $M \times M\text{-}N$ matrix whose columns are the M-N noise vectors, and $\mathbf{c}(\theta)$ is a direction vector. Note that $\mathbf{c}^{+}(\theta)\,E\,E^{+}\,\mathbf{c}(\theta)$ is the squared Euclidean distance from $\mathbf{c}(\theta)$ to the noise subspace. The noise subspace is the space spanned by the columns of the E matrix. A plot of q(θ) Vs θ exhibits peaks at those values of θ corresponding to the angles-of-arrival.

5) Once the angles-of-arrival have been estimated, the $\mathbf{B}$ matrix becomes available and can be used to determine the signal covariance matrix R_a as follows:

$$R_a = (\mathbf{B}^{+}\,\mathbf{B})^{-1}\,\mathbf{B}^{+}\,(\hat{R} - \sigma^2 I)\,\mathbf{B}\,(\mathbf{B}^{+}\,\mathbf{B})^{-1} \tag{12}$$

IV. PROPOSED TRACKING ALGORITHM

Let us now assume that the targets are moving and that during the interval $nT \le t \le nT + T$, the direction matrix $\mathbf{B}(t)$ essentially remains constant. Also assume that over the entire tracking interval the signal powers and their cross correlations do not change. Hence the source covariance matrix $R_a(nT)$ will be written as R_a. Then the sensor covariance equation (8) discretized at the time nT becomes:

$$R(nT) = \mathbf{B}(nT)\, R_a\, \mathbf{B}^{+}(nT) + \sigma^2 I, \quad n=1, 2, \ldots \tag{13}$$

In (13) $R(nT)$ is estimated using (8).

The tracking problem can be stated as follows: For $n=1,2 \ldots$, determine $\hat{R}(nT)$ from (8) and then use (13) to obtain an estimate of $\mathbf{B}(nT)$ with the columns correctly associated with $\mathbf{B}(nT-T)$. Suppose we start tracking at time T, and the MUSIC algorithm gives us an estimate of N, R_a, σ^2 and $\mathbf{B}(T)$. Let us rewrite (13) as follows:

$$R(nT) - \sigma^2 I = \mathbf{B}(nT)\, R_a\, \mathbf{B}^{+}(nT) \tag{14}$$

We now define an objective function as follows:

$$J\{\theta(nT)\} = \|\hat{R}(nT) - \mathbf{B}(nT)\, R_a \mathbf{B}^{+}(nT) - \sigma^2 I\|_F^2 \tag{15}$$

where $\|.\|_F^2$ denotes the Frobenius norm defined by $\|M\|_F^2 = \sum_j \sum_i |m_{ij}|^2$ where m_{ij} is the $(i,j)^{th}$ element of M.

In (15), $\hat{\mathrm{R}}(nT)$ is estimated from the data and the columns of $\mathbf{B}(nT)$ are functions of the unknown direction vectors $\theta(\mathrm{nT})$. Therefore by minimizing $J\{\theta(nT)\}$ over the columns of $\mathbf{B}(nT)$ we should get the best estimates of $\theta_i(nT)$, $i=1,\ldots N$, in the sense of minimizing the error between both sides of (14). Because of the nonquadratic nature of $J\{\theta(nT)\}$ and also because of the various permutations that are possible in the columns of $\mathrm{B}(nT)$, (15) will have many local minima. Thus if numerical iterative techniques are used to reach a minimum, the choice of an initial estimate will greatly influence which one of the minima is reached. We use the columns of $\mathbf{B}(nT - T)$ as the initial guess in the minimization algorithm. We repeat this procedure to obtain $\mathbf{B}(nT+T)$, with the columns of $\mathbf{B}(nT)$ as an initial estimate and so on. Note that the very first estimate $\mathbf{B}(T)$ is provided by MUSIC. In this way we are able to maintain correlation between the columns of the direction matrices for successive measurement intervals. Therefore tracking can be achieved.

Assuming that the elements of $\hat{R}(nT)$ are $x_{ij}+jy_{ij}$, and the elements of R_a are $a_{ij}+jb_{ij}$, extensive algebraic manipulation of (15) yields an expression of $J\{\theta(nT)\}$ given by:

$$J\{\theta(nT)\} = \sum_{q=1}^{M} \sum_{p=1}^{M} \{C_{pq}^2(nT) + D_{pq}^2(nT)\} \tag{16}$$

where

$$C_{pq}(nT) = x_{pq}(nT) - T_1 - T_2 - \sigma^2 \delta_{pq} \tag{17}$$

$$D_{pq}(nT) = y_{pq}(nT) - T_4 - T_3 \tag{18}$$

and δ_{pq} is the Kronecker delta. The terms T_1, T_2, T_3 and T_4 in (17) and (18) are functions of the angles-of-arrival and the elements of the source covariance matrix. By making use of the Vandermonde structure of the angles-of-arrival matrix $\mathbf{B}$ these terms are given by:

$$T_1 = \sum_{m=1}^{N} \sum_{l=1}^{N} a_{lm} \cos \{ (q-1)\, \mu_m(nT) - (p-1)\, \mu_l(nT) \} \tag{19}$$

$$T_2 = -\sum_{m=1}^{N} \sum_{l=1}^{N} b_{lm} \sin \{ (q-1)\, \mu_m(nT) - (p-1)\, \mu_l(nT) \} \tag{20}$$

$$T_3 = \sum_{m=1}^{N} \sum_{l=1}^{N} a_{lm} \sin \{ (q-1)\, \mu_m(nT) - (p-1)\, \mu_l(nT) \} \tag{21}$$

$$T_4 = \sum_{m=1}^{N} \sum_{l=1}^{N} b_{lm} \cos \{ (q-1)\, \mu_m(nT) - (p-1)\, \mu_l(nT) \} \tag{22}$$

An expression for the the gradient vector of $J\{\theta(nT)\}$ with respect to the vector $\theta(nT)$ can also be derived as follows:

$$\frac{\partial J}{\partial \theta_k(nT)} = 2\pi \frac{d}{\lambda} \cos\theta_k(t) \frac{\partial J}{\partial \mu_k(nT)} \quad \text{and}$$

$$\frac{\partial J}{\partial \mu_k(nT)} = \sum_{q=1}^{M} \sum_{p=1}^{M} \left\{ 2C_{pq}(nT) \frac{\partial C_{pq}}{\partial \mu_k(nT)} + 2D_{pq}(nT) \frac{\partial D_{pq}}{\partial \mu_k(nT)} \right\}; \quad k=1, \ldots N \tag{23}$$

where

$$\frac{\partial C_{pq}(nT)}{\partial \mu_k(nT)} = g_1 + g_2 + g_3 + g_4 \tag{24}$$

$$\frac{\partial D_{pq}(nT)}{\partial \mu_k(nT)} = e_1 + e_2 + e_3 + e_4 \tag{25}$$

and where

$$g_1 = (q-1) \sum_{i=1}^{N} a_{lk} \sin \{(q-1)\, \mu_k(nT) - (p-1)\mu_l(nT)\} \tag{26}$$

$$g_2 = -(q-1) \sum_{i=1}^{N} b_{lk} \cos \{(q-1)\, \mu_k(nT) - (p-1)\mu_l(nT)\} \tag{27}$$

$$g_3 = (p-1) \sum_{i=1}^{N} b_{kl} \cos \{(q-1)\, \mu_l(nT) - (p-1)\mu_k(nT)\} \tag{28}$$

$$g_4 = -(p-1) \sum_{i=1}^{N} a_{kl} \sin \{(q-1)\, \mu_l(nT) - (p-1)\mu_k(nT)\} \tag{29}$$

$$e_1 = (q-1) \sum_{i=1}^{N} b_{lk} \sin \{(q-1)\, \mu_k(nT) - (p-1)\mu_l(nT)\} \tag{30}$$

$$e_2 = -(q-1) \sum_{i=1}^{N} a_{lk} \cos \{(q-1)\, \mu_k(nT) - (p-1)\mu_l(nT)\} \tag{31}$$

$$e_3 = (p-1) \sum_{i=1}^{N} a_{kl} \cos \{(q-1)\, \mu_l(nT) - (p-1)\mu_k(nT)\} \tag{32}$$

$$e_4 = (p-1) \sum_{i=1}^{N} b_{kl} \sin \{(q-1)\, \mu_l(nT) - (p-1)\mu_k(nT)\} \tag{33}$$

Details of these derivations can be found in [9].

We can now summarize our tracking algorithm for a general setting, which consists of an arbitary number of targets emitting signals that may be correlated. We update the direction matrix every T seconds, and within each T second interval we use Q snapshots to estimate the sensor covariance matrix $\hat{R}(nT)$. The algorithm consists of the following steps:

1) At t = T, using Q snapshots from the array output, estimate the sensor covariance matrix $\hat{R}(T)$ using equation (8).

2) To this data apply the MUSIC algorithm discussed in section III. This gives the number of targets N, the source covariance matrix R_a, the sensor noise power σ^2 and estimates of the angular locations $\theta(T)$ of the targets at time T. Assume that the source covariance matrix remains unchanged.

3) At $t = T$ assume that the targets move to new positions, in the neighborhood of their old positions. Using another set of Q snapshots estimate the sensor covariance matrix $\hat{R}(2T)$.

4) Using the entries of the sensor covariance matrix $\hat{R}(2T)$, the entries of the source covariance matrix R_a, the sensor noise power σ^2, and N set up the objective function $J\{\theta(2T)\}$ as in (16) with the new angles-of-arrival $\theta(2T)$ as variables.

5) Determine the gradient vector of the minimizing function $J\{\theta(2T)\}$ with respect to $\theta(2T)$ using equations (23) to (33).

6) Equipped with the gradient vector, implement a gradient based minimization algorithm, such as the conjugate gradient method [10] to minimize $J\{\theta(2T)\}$. This algorithm is available in the IMSL subroutine library. Use $\theta(T)$ estimated in step 3 as the initial guess for the minimization. Upon convergence of the iterated estimate we get the new angles-of-arrival vector $\theta(2T)$ whose elements will be correctly associated with those of $\theta(T)$.

7) Repeat step 3 onwards for $3T$, $4T$ ....., to achieve tracking.

We summarize some of the features of our tracking algorithm by the following three propositions whose proofs can be found in [9].

PROPOSITION 1: A necessary condition for $J\{\theta(nT)\}$ to have a unique local minimum is :

$$E[\, a_i(t)\, a_i^+(t)\,] \neq E[\, a_j(t)\, a_j^+(t)\,] \quad \text{for } i \neq j \tag{34}$$

It is easy to see that if any two target signals have the same signal power then a permutation in the columns of the direction matrix B does not change the value of the objective function. Hence the minimum is not unique if condition (34) is violated.

PROPOSITION 2: When the source covariance matrix is diagonal, the value of $\theta(nT)$ that minimizes $J\{\theta(nT)\}$ is independent of the noise power σ^2.

The proof of this proposition is based on the property that the gradient of the Frobenius norm is independent of the noise power and consequently the estimate of the angles-of-arrival is independent of the noise power. However it is important to note that when tracking the angles-of-arrival based on an estimate of the sensor covariance matrix the accuracy of the converged estimate depends on the accuracy to which the sensor covariance matrix can be estimated, and this depends on the noise power σ^2.

PROPOSITION 3: In the case of two sensors and one target it can be shown that if $\lim_{Q\to\infty} \hat{R}(nT) = R(nT)$ then $\lim_{Q\to\infty} \hat{\theta}(nT) = \theta(nT)$.

By proposition 3, complete knowledge of the statistics of the target signals enables us to track the target perfectly.

V. SIMULATION EXAMPLE

In this example we consider the tracking of three targets with signal powers of 25, 16 and 36, using six sensors (M=6 and N=3). The targets move along parabolic trajectories in the directions shown in figure(2). The tracking algorithm is used to estimate the trajectories of the targets at a sensor signal to noise ratio of 5dB. Initial estimates of the number of targets (N), the sensor noise power (σ^2) and the angles-of-arrival are obtained from the MUSIC algorithm. A plot of $q(\theta)$ Vs θ as explained in step 4 of the MUSIC algorithm is shown in figure(3). The tracked trajectories are shown in figure(5) with an updating of the angles-of-arrival done with a $T = 2$ second interval and with $Q = 500$ snapshots within each interval.

Thus we see that when properly initiated the tracking algorithm follows closely the actual trajectories. The main feature of this approach is the capability of associating updated estimates with the previous estimates is embedded in the estimation procedure itself. Another feature is the simplicity of its implementation. This is due to the fact that the gradient expression has an analytical form with a well defined structure and hence there is no need to estimate it numerically.

VI. REFERENCES

1. B. Friedlander, *Optimum steady state position and velocity estimation using noisy sampled position data*, IEEE. Trans. Aerospace and electronics systems,Vol. AES-9, 1973, pp. 906-911.

2. P. Smith and G. Buechler, *A branching algorithm for discriminating and detecting multiple objects*, IEEE Trans. Automatic control,Vol. AC-20, 1975, pp. 101-104.

3. C. L. Morefield, *Application of 0-1 integer programming to multi-target tracking problems.*, Proc. IEEE Trans. Automatic Control,Vol. AC-22, 1977, pp. 301-312.

4. Y. Bar Shalom and E. Tse, *tracking in a cluttered environment with probabilistic data association*, Automatica,Vol. 11, 1975, pp. 451-460.

5. R. A. Singer, R .G. Sea and K. Housewright, *Derivation and evaluation of improved tracking filters for use in dense multitarget environments*, IEEE Trans. Inform. Theory,Vol. IT-20, 1974, pp. 423-432.

6. R. O. Schmidt, *Multiple emitter location and signal parameter estimation*, IEEE. Trans. Antennas and Propagation,Vol. AP-34, 1986, pp. 276-280.

7. R.O. Schmidt, A signal subspace approach to multiple source location and spectral estimation., PhD dissertation, Stanford University, CA, 1981.

8. S. Haykin et al., Eds, Array Signal Processing, Englewood Cliffs, NJ, Prentice hall, 1985.

9. C. R. Sastry, Multiple Target Tracking Using a Matrix Norm Minimization Technique, Master's thesis, Department of Electrical Engineering, and Signal Processing and Interpretation Laboratory Report # SPIL-88-04, University of Pittsburgh, Pittsburgh, PA, Aug 1988.

10. D. G. Luenberger, Introduction to Linear and Nonlinear Programming, Addison-Wesley, 1973.

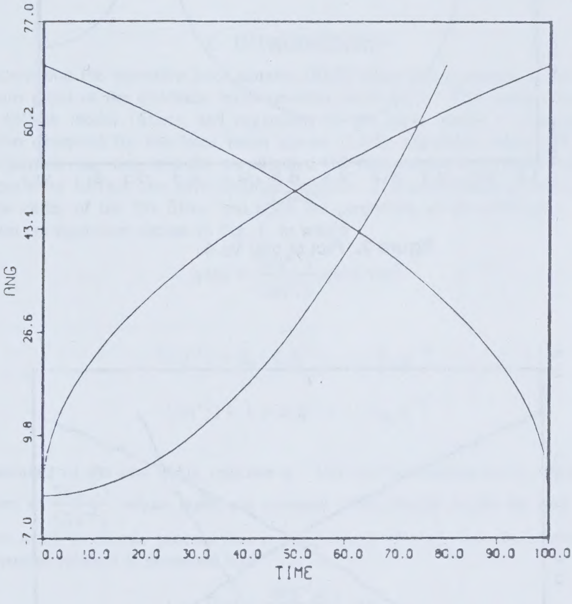


figure 2. Actual trajectory of the three targets.

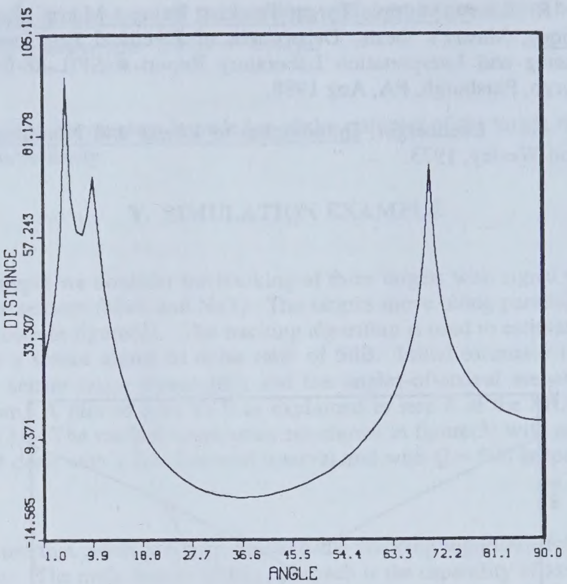


figure 3. Plot of q(θ) Vs θ.

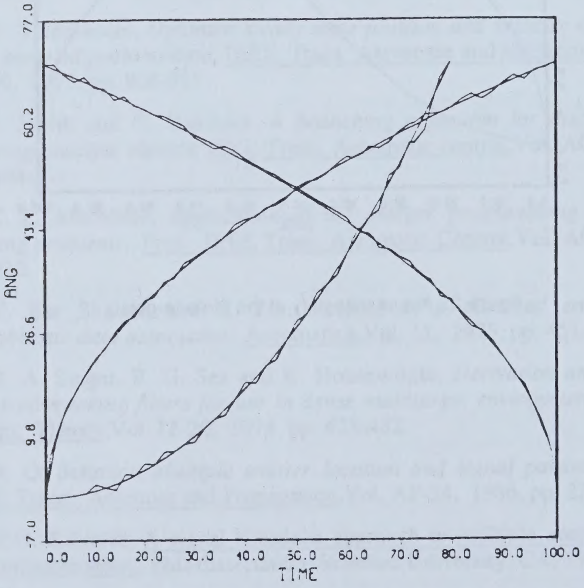


figure 4. Tracked trajectory of the three targets.

INSUFFICIENT ORDER ADAPTIVE IIR FILTERS WITH WHITE INPUTS

MAJID NAYERI
Michigan State University, East Lansing, Michigan

ABSTRACT

The definition of degenerated points is generalized for adaptive IIR filters. Using this definition, sufficient conditions for multimodality of the error surfaces associated with a class of "insufficient order" adaptive filters with white input are obtained.

1. INTRODUCTION

It is well-known that the recursive least squares (RLS) algorithm possesses a unique solution which is the unique minimum point of the quadratic equation-error surface [1]. This uniqueness holds regardless of the order of the ARMA model (filter), and regardless of the input which is persistently exciting. This characteristic is also observed for the least mean square (LMS) algorithm when FIR filters are used [2]. However, this uniqueness may not hold for an adaptive IIR filter which minimizes the mean-square output error, i.e., the output-error surface can have multiple minima. The multiplicity of minimum points is highly dependent upon the order of the IIR filter, and upon the properties of the rich input. Let us consider the system identification configuration shown in Fig. 1, in which

$$y(n) = \frac{D(q^{-1})}{C(q^{-1})}x(n) + v(n) \tag{1.1}$$

where

$$D(q^{-1}) = d_0 + d_1 q^{-1} + \ldots + d_{n_d} q^{-n_d},$$

$$C(q^{-1}) = 1 + c_1 q^{-1} + \ldots + c_{n_c} q^{-n_c}$$

are coprime polynomials of the unit delay operator q^{-1} and $v(n)$ is additive noise. The transfer function of the unknown system is $\frac{D(z^{-1})}{C(z^{-1})}$ whose poles are assumed to lie strictly inside the unit circle. The additive noise $v(n)$ is a zero mean stochastic process that is independent of $x(n)$. Let the adaptive system be an IIR filter whose input-output relation is governed by

$$\hat{y}(n) = \frac{B(q^{-1},n)}{A(q^{-1},n)}\, x(n), \tag{1.2}$$

where

$$B(q^{-1},n) = b_0(n) + b_1(n)\, q^{-1} + \ldots + b_{n_b}(n)\, q^{-n_b},$$

$$A(q^{-1},n) = 1 + a_1(n)\, q^{-1} + \ldots + a_{n_a}(n)\, q^{-n_a}$$

Then the error surfaces of the adaptive IIR filters can be classified [3] as:

1) Sufficient order filter[1] with white noise excitation;
2) Sufficient order filter with colored (noise) excitation;
3) Insufficient order filter with white noise excitation;
4) Insufficient order filter with colored (noise) excitation;

The sufficient conditions for unimodality of the classes 1) and 2) are given in [4]. The existence of multiple minima for class 1), which was conjectured to be unimodal according to Stearns [5], has been shown in [6,7].

---

[1] A filter is of sufficient order if $n_a \geq n_c$ and $n_b \geq n_d$.

Here, we present some analysis along with some examples to analyze the class 3). The problem formulation and the main theory are introduced in section 2. The development of the theory involves: (i) generalization of the *degenerated* points introduced in [8], (ii) recognition of the connection between the existence of such points and multimodality of class 3). In section 3, class 3) error surfaces are further considered and sufficient conditions for multimodality of a subclass of 3) are introduced.

2. DEGENERATED POINTS AND SADDLE POINTS

Let us consider the model representation in Fig. 1 where the adaptive IIR filter in (1.2) is considered frozen. In other words, it does not depend on the time variable n. Then, the mean square error (MSE) is given by

$$E[e^2(n)] = E\left[\left[\frac{D(q^{-1})}{C(q^{-1})} - \frac{B(q^{-1})}{A(q^{-1})}\right]x(n)\right]^2 + E[v^2], \tag{2.1}$$

which is a function of the parameter vector $\theta(\underline{a},\underline{b})$ where

$$\underline{a} = [a_1 \cdots a_{n_a}],$$

$$\underline{b} = [b_0\, b_1 \cdots b_{n_b}].$$

The stationary points of the MSE functional are the solutions of

$$E\left[\left[\frac{D(q^{-1})}{C(q^{-1})} - \frac{B(q^{-1})}{A(q^{-1})}\right]x(n) \cdot \frac{B(q^{-1})}{A^2(q^{-1})}\,x(n-i)\right] = 0\,, \quad 1 \le i \le n_a, \tag{2.2a}$$

$$E\left[\left[\frac{D(q^{-1})}{C(q^{-1})} - \frac{B(q^{-1})}{A(q^{-1})}\right]x(n) \cdot \frac{1}{A(q^{-1})}\,x(n-j)\right] = 0\,, \quad 0 \le j \le n_b, \tag{2.2b}$$

which are obtained by setting to zero the gradient of $E[e^2(n)]$ with respect to $\underline{a}$ and $\underline{b}$. Let us assume that θ^* is a stationary point and correspondingly $A^*(q^{-1})$ and $B^*(q^{-1})$ satisfy (2.2a) and (2.2b). Then, it can be shown that the MSE incurred at θ^* is given by

$$E[e^2(n)] = E\left[\frac{D(q^{-1})}{C(q^{-1})}x(n)\right]^2 - E\left[\frac{B^*(q^{-1})}{A^*(q^{-1})}x(n)\right]^2 + E[v^2]. \tag{2.3}$$

Note the close similarity of (2.3) with the minimum norm obtained by the minimum norm estimator in linear spaces. As a matter of fact, if an adaptive FIR filter is used, i.e., when $A(q^{-1}) = 1$, there is only one stationary point, which corresponds to the Wiener solution of the underlying linear vector space.

Soderstrom [8] introduced the notion of *degenerated points* for adaptive filters, i.e., those points such that $B(q^{-1}) \equiv 0$. Since we intend to use the more general filter structure (1.2), a more general definition of degenerated points is introduced. Based on this definition, it will be shown that a degenerated stable stationary point is a saddle point. Existence of such a saddle point which is equivalent to the existence of multiple minima is the key to establishing a set of sufficient conditions which result in the multimodality of error surfaces when x(n) is white.

Definition:

The degenerated points are the points in the parameter space such that $B(q^{-1}) \equiv 0$ if $n_b < n_a$, and $B(q^{-1}) \equiv L(q^{-1})A(q^{-1})$ if $n_b \ge n_a$ where $L(q^{-1}) = l_0 + l_1 q^{-1} + \cdots + l_{n_b-n_a} q^{-(n_b-n_a)}$, which includes $B(q^{-1}) \equiv 0$.

Throughout the remainder of this section x(n) is assumed to be white, although some of the following results may also apply to colored x(n).

Theorem 1:

A stable degenerated solution of (2.2) is a saddle point.

Proof:

Based on the definition of degenerated points we consider two cases:

Case 1: $n_b<n_a$

Since $B(q^{-1})\equiv 0$, (2.2) is reduced to

$$E\left[\frac{D(q^{-1})}{C(q^{-1})}x(n)\cdot\frac{1}{A(q^{-1})}x(n-j)\right]=0\ ,\ 0\leq j\leq n_b, \tag{2.4}$$

It has been proven in [8] that the solutions of (2.4) are saddle points.

Case 2: $n_b\geq n_a$

The degenerated points are of the form $B(q^{-1})=L(q^{-1})A(q^{-1})$ and (2.2) is reduced to

$$E\left[\left[\frac{D(q^{-1})}{C(q^{-1})}-L(q^{-1})\right]x(n)\cdot\frac{1}{A(q^{-1})}x(n-j)\right]=0\ ,\ 0\leq j\leq n_b\,. \tag{2.5}$$

Now it can be shown that stable solutions of (2.5) are saddle points. If the polynomial $B(q^{-1})$ is represented as

$$B(q^{-1})=L(q^{-1})\cdot A(q^{-1})+\beta(q^{-1})$$

where

$$\beta(q^{-1})=\sum_{i=n_b-n_a+1}^{n_b}\beta_i q^{-i},$$

then minimizing (2.1) is equivalent to minimizing

$$E\left[\left[\frac{D(q^{-1})}{C(q^{-1})}-L(q^{-1})-\frac{\beta(q^{-1})}{A(q^{-1})}\right]x(n)\right]^2 \tag{2.6}$$

with respect to the parameters a_i , l_j and β_k . This is so because there is a one-to-one correspondence between ($\underline{a}$, $\underline{b}$) and ($\underline{a}$, $\underline{l}$, $\underline{\beta}$) points. If the gradient of (2.6) with respect to l_j parameter is set to zero, we will have

$$E\left[\left[\frac{D(q^{-1})}{C(q^{-1})}-L(q^{-1})-\frac{\beta(q^{-1})}{A(q^{-1})}\right]x(n)\cdot x(n-j)\right]=0\ ,\ \ 0\leq j\leq n_b-n_a. \tag{2.7}$$

Note that

$$E\left[\frac{\beta(q^{-1})}{A(q^{-1})}x(n)\cdot x(n-j)\right]=0\ ,\ \ 0\leq j\leq n_b-n_a, \tag{2.8}$$

since x(n) is white. This clearly shows that the optimal $L(q^{-1})$ polynomial, denoted by $L^*(q^{-1})$, is the same for all the stationary points of (2.5) and satisfies

$$E\left[\left[\frac{D(q^{-1})}{C(q^{-1})}-L^*(q^{-1})\right]x(n)\cdot x(n-j)\right]=0\ ,\ \ 0\leq j\leq n_b-n_a. \tag{2.9}$$

If $L^*(q^{-1})$ is substituted in (2.6), $\left[\dfrac{D(q^{-1})}{C(q^{-1})} - L^*(q^{-1})\right]$ is then fixed. The degenerated point solutions corresponding to $\beta(q^{-1}) \equiv 0$ are the solutions of

$$E\left[\frac{1}{A^*(q^{-1})}x(n-j)\cdot\left[\frac{D(q^{-1})}{C(q^{-1})} - L^*(q^{-1})\right]x(n)\right] = 0\,, \quad n_b - n_a + 1 \leq j \leq n_b, \tag{2.10}$$

which corresponds to a saddle point by [8]. □

This theorem simply states that the degenerated points are not extreme points. However, the significance of these points is expressed by the next theorem which predicts the multimodality of the error surface if such points exist. In other words, non-uniqueness of the estimates could be concluded.

Theorem 2:

If $n_a > n_b = 0$ or $n_a = 1$, the existence of a stable degenerated solution of (2.2) implies the multimodality of the error surface of (2.1).

Proof:

Assume $n_a > n_b = 0$, we can express $E[e^2(n)]$ as

$$W(\underline{a}, \underline{b}) \triangleq E[e^2(n)] = P_0 - 2P_1(\underline{a})^T\underline{b} + \underline{b}^T P_2(\underline{a})\underline{b} \tag{2.11}$$

where

$$P_0 = E\left[\frac{D(q^{-1})}{C(q^{-1})}x(n)\right]^2$$

$$P_1(\underline{a})_j = E\left[\frac{D(q-1)}{C(q^{-1})}x(n)\cdot\frac{1}{A(q^{-1})}x(n-j)\right] \qquad 0 \leq j \leq n_b \tag{2.12}$$

$$P_2(\underline{a})_{jk} = E\left\{\left[\frac{1}{A(q^{-1})}x(n-j)\right]\left[\frac{1}{A(q^{-1})}x(n-k)\right]\right\} \qquad 0 \leq j,k \leq n_b.$$

Now, since $P_2(\underline{a})$ is always a positive definite matrix for sufficiently rich input $x(n)$, we could optimize $W(\underline{a}, \underline{b})$ with respect to $\underline{b}$ by choosing

$$\underline{b}^0 = P_2(\underline{a})^{-1}\,P_1(\underline{a}). \tag{2.13}$$

The function $W(\underline{a},\underline{b}^0)$ is the reduced order error surface in n_a parameters. Since $n_b = 0$ the set of stable degenerated point solutions (2.4) represents an n_a dimensional surface, $P_1(\underline{a}) = 0$,in the n_a dimensional space $\underline{a}$ which separates the stability region, i.e., it partitions the stability region. The MSE value P_0 for the stable degenerated points $(\underline{a}^*, \underline{0})$ is the same as the MSE value for the boundary points of the stability region in this case (see the Appendix). Now since a stable degenerated solution is a saddle point by Theorem 1, any direction away from the stable degenerated point surface decreases the MSE as seen by

$$W(\underline{a}^* + \delta\underline{a}, \underline{b}^0(\underline{a}^* + \delta\underline{a})) = P_0 - P_1(\underline{a}^* + \delta\underline{a})^T\,P_2^{-1}(\underline{a}^* + \delta\underline{a})\,P_1(\underline{a}^* + \delta\underline{a}) < P_0, \tag{2.14}$$

This means that in every partition there is a minimum point. In other words, there exist multiple minima.

For $n_a = 1$ and $n_b \geq 1$ we have $\beta(q^{-1}) = \beta_{n_b}q^{-n_b}$ and $\dfrac{B(q^{-1})}{A(q^{-1})} = L(q^{-1}) + \dfrac{\beta_{n_b}q^{-n_b}}{A(q^{-1})}$, for which the proof is immediate by the above argument. □

It should be pointed out that the analysis of degenerated solutions is independent of the parameter $n^* \triangleq \min(n_a-n_c, n_b-n_d)$. This means that Theorem 1 and Theorem 2 are true for both sufficient order case, i.e., when $n^* \geq 0$, and insufficient order case, $n^* < 0$. In [6-8] examples were shown for sufficient order cases. We now concentrate on insufficient order class 3).

3. INSUFFICIENT ORDER FILTERS

As stated before, Theorem 2 can also be applied to insufficient order filters where $n^*<0$. It provides the necessary tool to establish conditions for existence of multiple minima in these cases. In order to establish a systematic method to find multiple minima on MSE surfaces, we can exploit Theorem 2 even further in obtaining conditions, with respect to the unknown system characteristics, which guarantee the existence of multiple minima. These conditions will then be used to demonstrate that some examples, for the insufficient order case, can be studied in a yet simpler way.

Theorem 3:

Consider the system (1.1) in which x(n) is white and $n^*<0$.

1. When $n_b=n_a-1$, stable degenerated solutions exist if $D(z^{-1})$ has at least (n_b+1) zeros outside the unit circle. For the case in which $n_a = 1$ and $n_b = 0$, this means that the error surface has multiple minima.
2. When $n_b \geq n_a$, let $L^*(z^{-1})$ satisfy (2.9). Then, stable degenerated solutions exist if $[D(z^{-1})-L^*(z^{-1})C(z^{-1})]$ has at least (n_b+1) zeros outside the unit circle. For the case in which $n_a = 1$, this means that the error surface has multiple minima.

Proof:

Equation (2.5) at the solutions $A^*(z^{-1})$ and $L^*(z^{-1})$ could be rewritten as

$$\begin{aligned} 0 &= \frac{1}{2\pi i}\oint_{|z|=1} \frac{1}{A^*(z^{-1})} \left[\frac{D(z)}{C(z)} - L^*(z)\right] z^{-j}\,\frac{dz}{z}\,, \quad 0 \leq j \leq n_b \\ &= \frac{1}{2\pi i}\oint_{|z|=1} \frac{1}{z^{n_a}A^*(z^{-1})}\,\frac{[D(z)-L^*(z)C(z)]}{C(z)}\,\frac{z^{\max(0,p)}}{z^{\max(0,-p)}}\,z^{n_b-j}\,dz, \quad 0 \leq j \leq n_b \end{aligned} \tag{3.1}$$

where $p=n_a-n_b-1$.

Note that (2.4) is embedded in (3.1). Now, we use the result of Astrom and Soderstrom [9] which is of central importance in analyzing Eq. (3.1).

Lemma 1:[9]

Let $f(z) = \dfrac{g(z)}{h(z)}$, where g(z) is analytic inside and on the unit circle and h(z) is a polynomial of degree n with all zeros strictly inside the unit circle. Assume that

$$\frac{1}{2\pi i}\oint_{|z|=1} f(z) z^k \, dz = 0, \quad 0 \leq k \leq (n-1).$$

Then *all* zeros of h(z) are also zeros of g(z), i.e., f(z) is analytic inside the unit circle. □

From Equation (3.1), it is observed that

$$\begin{aligned} \text{number of poles} &= n_a + \max(0, n_b-n_a+1) \\ &= \max(n_a, n_b+1) \end{aligned}$$

$$\text{number of equations} = (n_b+1)$$

Since the number of poles should be less than or equal to the number of equations if we wish to apply Lemma 1, then we require

$$\boxed{(n_b + 1) \geq n_a}$$

We now distinguish two cases:

1. **$n_b + 1 = n_a$**

   As discussed before, $B^*(q^{-1}) \equiv 0$ or equivalently $L^*(q^{-1}) \equiv 0$ for this case. Thus $\left[\frac{1}{z^{n_a}A^*(z^{-1})} \cdot \frac{D(z)}{C(z)}\right]$, in Equation (3.1), should be analytic inside the unit circle. This can occur only when D(z) has at least n_a zeros inside the unit circle so that the cancelled poles are stable ones. Thus we conclude that if $D(z^{-1})$ has at least n_a zeros *outside* the unit circle, stable degenerated solutions exist. Note that this excludes $n^* \geq 0$, in which case, $D(z^{-1})$ can not have n_a zeros because $n_a > n_b \geq n_d$. Exclusion of sufficient order filters is further justified by Soderstrom's unimodality condition [4] being automatically satisfied in this case since $(n_b+1) = n_a \geq n_c$. Existence of a stable degenerated solution in the special case in which $n_a = 1$ and $n_b = 0$ implies multimodality of the error surface according to Theorem 2.

2. **$n_b + 1 > n_a$**

   Note that $L^*(q^{-1})$ is not necessarily zero and its solution is found by (2.9). But (3.1) is reduced to

   $$\frac{1}{2\pi i} \oint_{|z|=1} \frac{1}{z^{(n_b-n_a+1)}\, z^{n_a}A^*(z^{-1})} \cdot \frac{[D(z)-L^*(z)C(z)]}{C(z)}\, z^{n_b-j}dz=0\ ,\ 0 \leq j \leq n_b\,. \tag{3.2}$$

   We then conclude that $[D(z)-L^*(z)C(z)]$ must have an (n_b-n_a+1)th order zero at $z=0$ and that the sufficient condition for the existence of a stable $A^*(z^{-1})$ is that among the remaining zeros of $D(z)-L^*(z)C(z)$ at least n_a of them are inside the unit circle. Note that since $\deg\,(D(z)-L^*(z)C(z))=n_b-n_a-n^*$ and since $n^*<0$, the degree requirement is automatically satisfied. Application of Theorem 2 implies multimodality of the error surface when $n_a = 1$. □

Example 1:

Let us use the adaptive filter $\frac{b}{1-az^{-1}}$ to find the MMSE estimate of the stable unknown system $\frac{d_0+d_1z^{-1}}{1+c_1z^{-1}+c_2z^{-2}}$. Since $n_b = 0$ and $n_a = 1$, by Theorem 3, the function

$$\left[\frac{1}{z-a^*} \cdot \frac{d_0+d_1z}{1+c_1z+c_2z^2}\right] \tag{E1.1}$$

has to be analytic inside the unit circle. Let $d_1 \neq 0$. If $D(z^{-1}) = d_0 + d_1z^{-1}$ has its zero *outside* the unit circle, i.e., $|\frac{d_0}{d_1}| < 1$, then $a^* = -\frac{d_0}{d_1}$ is a stable degenerated solution. This guarantees the existence of multiple minima. For instance, if the unknown system is given by $\frac{0.05-0.4z^{-1}}{1-1.131z^{-1}+0.25z^{-2}}$, Fig. 1 shows the normalized error surface which already appeared in [10]. On the other hand, if $d_1=0$, then (E1.1) is analytic only if $|a^*| >1$ and no stable degenerated point exists. The unimodal characteristic of the error surface when the unknown system is given by $\frac{0.05}{1-1.75z^{-1}+0.81z^{-2}}$ was shown in [10]. □

Example 2:

Suppose that the unknown system in Fig. 1 is given by $d_0+d_1z^{-1}+d_2z^{-2}$ and an adaptive IIR filter of the form $\frac{b_0+b_1z^{-1}}{1-az^{-1}}$ is used to minimize the mean square error. From the identity $B^*(z^{-1})=L^*(z^{-1})A^*(z^{-1})$, we find that $L^*(z^{-1})=b_0^*$ and $a^*b_0^*+b_1^*=0$. Since $D(z)-L^*(z)C(z)=(d_0-b_0^*)+d_1z+d_2z^2$ has a first-order zero at $z=0$, it is concluded that $b_0^*=d_0$. (Consistent with Eq. (2.9).) Also, $(d_1z+d_2z^2)$ has a nonzero root inside the unit circle if $|\frac{d_1}{d_2}|<1$ which implies that $\frac{d_1z+d_2z^2}{z-a^*}$ is analytic inside the unit circle with $a^*=-\frac{d_1}{d_2}$ and $|a^*|<1$. Thus, multiple minima exist if $|\frac{d_1}{d_2}|<1$. This is consistent with the development in [5]. It is seen that if $|\frac{d_1}{d_2}|>1$, then no stable degenerated solution exists and [5] shows the unimodal characteristic of the error surface. □

From the examples presented here it may seem that the existence of stable degenerated solutions is a possible necessary condition for multimodality of error surfaces [11]. However, the following example rule out this possibility.

Example 3:

Consider example 1. It was concluded that when the unknown has its zero outside the unit circle, the error surface is multimodal. But, what if the unknown system is minimum phase, i.e., the zero is inside the unit circle? Let $c_2=0$, $c_1=0.99$, $d_0=1$ and $d_1=-0.85$. Figure 3 shows the error surface of the adaptive filter which is multimodal with no degenerated solutions. Therefore, the existence of the degenerated solutions is *not* a necessary condition. □

4. SUMMARY

Four classes of error surfaces, in accordance with the order of adaptive filters and their inputs, were introduced and the existence of local minima for some classes was discussed. Classes 1) and 2) have been shown to be multimodal in the literature [6-8]. In this paper Soderstrom definition of degenerated points was generalized, and applied to the more general class 3). The error surfaces associated with class 3) were investigated and, as Theorem 3 suggested that multimodality of some of these surfaces could be conveniently concluded when the unknown system is not minimum phase.

ACKNOWLEDGEMENT

The author wishes to thank H. Fan and W. K. Jenkins for their helpful suggestions in the presentation of the result.

REFERENCES

[1] L. Ljung and T. Soderstrom, *Theory and Practice of Recursive Identification*. Cambridge, MA: The MIT Press, 1983.

[2] B. Widrow, J. M. McCool, M. G. Larimore, and C. R. Johnson, Jr., "Stationary and nonstationary learning characteristics of the LMS adaptive filter," *Proc. IEEE*, vol. 64, no. 8, pp. 1151-1162, Aug. 1976.

[3] H. Fan and W. K. Jenkins, "A new adaptive IIR filter," *IEEE Trans. Circuit. Syst.*, vol. CAS-33, no. 10, pp. 939-947, Oct. 1986.

[4] T. Soderstrom and P. Stoica, "Some properties of the output error method," *Automatica*, vol. 18, no. 12, pp. 1692-1716, Dec. 1982.

[5] S. D. Stearns, "Error surfaces of recursive adaptive filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, no. 3, pp. 763-766, June 1981.

[6] H. Fan and M. Nayeri, "Counter examples to Stearns' conjecture on error surfaces of adaptive IIR filters," Companion paper, Submitted to *IEEE Trans. Acoust., Speech, Signal Processing*, 1987.

[7] H. Fan and M. Nayeri, "Counter examples to Stearns' conjecture on error surfaces of adaptive IIR filters," *Proc. IEEE Intl. Conf. Acoust., Speech, Signal Processing*, New York, NY, Apr. 11-14, 1988.

[8] T. Soderstrom, "On the uniqueness of maximum likelihood identification," *Automatica*, vol. 11, no. 2, pp. 193-197, 1975.

[9] K. J. Astrom and T. Soderstrom, "Uniqueness of the maximum likelihood estimation of the parameters of an ARMA model," *IEEE Trans. Automat. Contr.*, vol. AC-19, pp. 769-773, 1974.

[10] C. R. Johnson, Jr., and M. G. Larimore, "Comments on and additions to 'An adaptive recursive LMS filter'," *Proc. IEEE*, vol. 65, no. 9, pp. 1399-1402, Sept. 1977.

[11] M. Nayeri, "Improvements in adaptive IIR filtering theory and application'" PhD Dissertation, University of Illinois, Urbana, IL, May 1988.

APPENDIX

Corollary:

Let x(n) be a stationary process with the power spectrum $\Phi(e^{i\omega})$ such that $\Phi(e^{i\omega}) \neq 0$ for $\omega \varepsilon [-\pi,\pi]$. Then

$$E\left[\left|\frac{1}{A(q^{-1})}x(n)\right|^2\right] \to +\infty$$

if any zero of the stable polynomial $A(z^{-1})$ approaches the unit circle.

Proof:

We have that

$$E\left[\left|\frac{1}{A(q^{-1})}x(n)\right|^2\right] = \frac{1}{2\pi i}\oint_{|z|=1}\left|\frac{1}{A(z)}\right|^2 \Phi(z)\,\frac{dz}{z}$$

$$= \frac{1}{2\pi}\int_{-\pi}^{\pi}\left|\frac{1}{A(e^{i\omega})}\right|^2 \Phi(e^{i\omega})\,d\omega$$

Let us now assume that $A(z^{-1})$ is such that it has only one real zero. Then we can write

$$A(e^{i\omega}) = (1-re^{-i\omega})\,A_1(e^{i\omega})$$

where $A_1(e^{i\omega}) \neq 0$ for all ω. Then

$$I \triangleq \lim_{|r|\to 1} \frac{1}{2\pi}\int_{-\pi}^{\pi}\left|\frac{1}{A(e^{i\omega})}\right|^2 \Phi(e^{i\omega})\,d\omega = \begin{cases} \displaystyle\lim_{\alpha\to 0^+} \frac{1}{4\pi}\int_{\alpha}^{\pi}\frac{f(e^{i\omega})}{\sin^2(\frac{\omega}{2})}\,d\omega & \text{if } r=1 \\ \displaystyle\lim_{\alpha\to 0^+} \frac{1}{4\pi}\int_{0}^{\pi-\alpha}\frac{f(e^{i\omega})}{\cos^2(\frac{\omega}{2})}\,d\omega & \text{if } r=-1 \end{cases}$$

where $f(e^{i\omega}) \triangleq \left|\frac{1}{A_1(e^{i\omega})}\right|^2 \Phi(e^{i\omega})$ is real with the following properties:

(i) $f(e^{i\omega}) \neq 0$ for all ω
(ii) $f(e^{i\omega}) = f(e^{-i\omega})$

Note that (i) implies that $f(e^{i\omega})$ is bounded away from zero and therefore, there exists an $\varepsilon>0$ such that $f(e^{i\omega}) \geq \varepsilon$. Thus, when $r=1$ we have

$$I_\alpha \triangleq \frac{1}{4\pi}\int_\alpha^\pi \frac{f(e^{i\omega})}{\sin^2(\frac{\omega}{2})}\, d\omega \;\geq\; \frac{\varepsilon}{4\pi}\int_\alpha^\pi \frac{1}{\sin^2(\frac{\omega}{2})}\, d\omega \;\geq\; \frac{\varepsilon}{4\pi}\left|\int_\alpha^\pi \frac{\cos(\frac{\omega}{2})}{\sin^2(\frac{\omega}{2})}\, d\omega\right|$$

$$= \frac{\varepsilon}{2\pi}\left|\frac{1}{\sin(\frac{\alpha}{2})} - 1\right|$$

Similarly, for $r=-1$

$$I_\alpha \geq \frac{\varepsilon}{2\pi}\left|\frac{1}{\cos(\frac{\pi-\alpha}{2})} - 1\right|$$

Therefore, it is concluded that

$$I = \lim_{\alpha\to 0^+} I_\alpha = +\infty$$

Let us now assume that the polynomial $A(z^{-1})$ has only one pair of conjugate poles represented by $re^{i\theta}$ and $re^{-i\theta}$, $-\pi<\theta<\pi$. Therefore, it can be represented as

$$A(e^{i\omega}) = (1-re^{i(\theta-\omega)})(1-re^{-i(\theta+\omega)})\overline{A_1}(e^{i\omega})$$

Then, by letting $r=1$ and using the same argument as before

$$\overline{I} = \lim_{\alpha\to 0^-} \frac{1}{4\pi}\int_0^{\theta-\alpha} \frac{\overline{f}(e^{i\omega})}{\sin^2(\frac{\theta-\omega}{2})\,\sin^2(\frac{\theta+\omega}{2})}\, d\omega \;+\; \lim_{\alpha\to 0^+} \frac{1}{4\pi}\int_{\theta+\alpha}^{\pi} \frac{\overline{f}(e^{i\omega})}{\sin^2(\frac{\theta-\omega}{2})\,\sin^2(\frac{\theta+\omega}{2})}\, d\omega$$

$$\geq \overline{\varepsilon}\frac{M}{2\pi}\left\{\lim_{\alpha\to 0^-}\left|\frac{1}{\sin(\frac{\alpha}{2})} - \frac{1}{\sin(\frac{\theta}{2})}\right| + \lim_{\alpha\to 0^+}\left|\frac{1}{\sin(\frac{\alpha}{2})} - \frac{1}{\cos(\frac{\theta}{2})}\right|\right\}$$

where

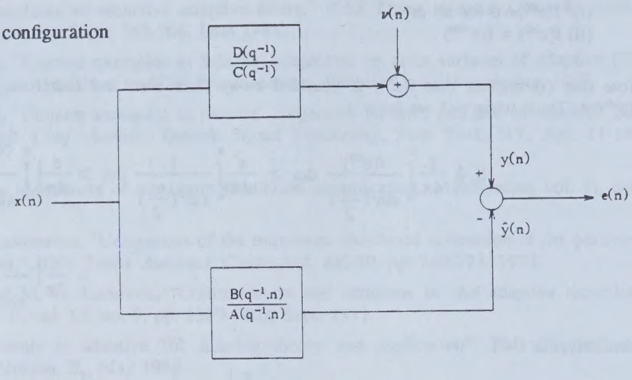
$$M \geq \frac{1}{\sin^2(\frac{\theta+\omega}{2})}$$

and where $\overline{f}$ and $\overline{\varepsilon}$ are appropriately defined by using $\overline{A_1}$ for A_1 above. Again it is obvious that $\overline{I} = +\infty$.

The case where $A(z^{-1})$ has more zeros on the unit circle can be treated similarly by evaluating the improper integral I in the same manner. □

It is important to note that in a more general, $\det(P_2(\underline{a})) \to +\infty$ as *any* zero of the stable polynomial $A(z^{-1})$ approaches the unit circle since $\mathrm{Trace}(P_2(\underline{a})) \to +\infty$. Equivalently, $P_2^{-1}(\underline{a})$ approaches *singularity.* But it is identically equal to the $(n_b+1)\mathrm{x}(n_b+1)$ zero matrix only when (n_b+1) zeros of $A(z^{-1})$ are located on the unit circle. Therefore, the claim made in [8] is not generally correct, i.e., the reduced error surface $W(\underline{a},\underline{b}^0)$ is not equal to P_0 every where on the boundaries of the stability region.

Fig. 1. System identification configuration



$\nu(n)$
$\frac{D(q^{-1})}{C(q^{-1})}$
+
$y(n)$
+
$x(n)$
$e(n)$
-
$\hat{y}(n)$
$\frac{B(q^{-1},n)}{A(q^{-1},n)}$

Fig. 2. Normalized error contours of Example 1. All the degenerated points are shown by the heavy line. The point (0.125,0) is the only degenerated solution.

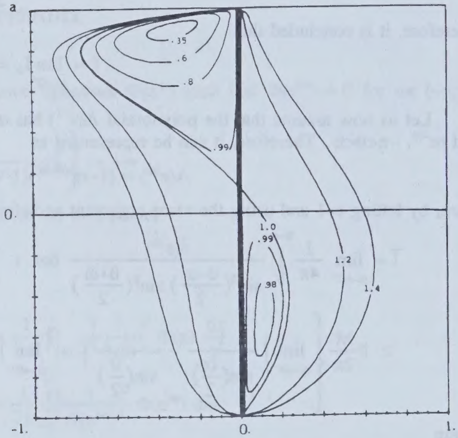
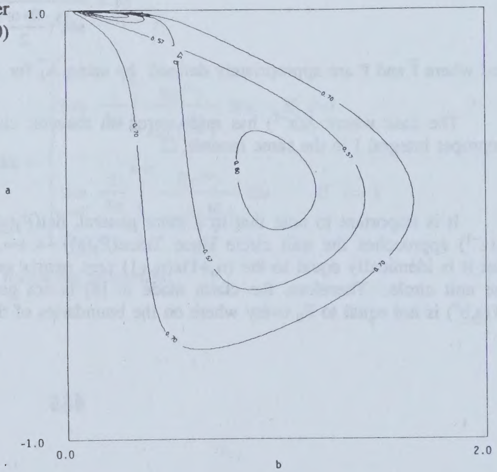


Fig. 3. The multimodal error surface of the filter in Example 3 with no degenerated point (b=0) solutions.



THE PERFORMANCE CHARACTERISTICS OF THE ORDER RECURSIVE ADAPTIVE LATTICE EQUALIZER

B. M. SALLAM, S. B. KESLER
Electrical and Computer Engineering Department,
Drexel University, Philadelphia, PA 19104

I-ABSTRACT

We investigate and compare the initial convergence and the steady state performance of two least squares lattice channel equalizers. The analyses are confirmed by simulation results.

II-INTRODUCTION

In this paper we address two issues in the least squares lattice (LSL) algorithms as they pertain to adaptive channel equalizers. First, we analyze the effects of the equalizer initialization procedures on the initial convergence speed and second, we compare the steady state mean square error (MSE) at the equalizer output with the theoretically attainable value.

The above analysis is performed using two different adaptive channel equalizer algorithms. The first is the conventional mixed time and order (MTO) recursive LSL, introduced by Satorius and Pack [1], and the other one is the order recursive adaptive lattice (ORAL) equalizer, recently introduced by authors [2] which is based on the generalized residual energies (GRE), [3]. The theoretical analysis of the initial convergence conditions supported by computer simulation, shows that the MTO recursive LSL convergence speed depends on the initial conditions, i.e., on the choice of the initial value of the channel output autocorrelation matrix (ACM), in other words smaller values enables faster convergence, however, too small a value slows down the convergence rate, due to finite word effects. The optimum weight of the adaptive least squares filter is given by

$$\mathbf{w}_{opt} = \mathbf{R}^{-1} \mathbf{q} \tag{1}$$

where **R** and **q** are the steady state ACM of the stationary channel output and the crosscorrelation vector (CCV) between the channel output and the desired sequences. (channel input), respectively.

During the training mode, the equalizer adjusts its coefficients to match those of the unknown channel. Therefore the estimated coefficient vector is,

$$\mathbf{w}(n)=\hat{\mathbf{R}}(n)^{-1}\,\hat{\mathbf{q}}(n) \tag{2}$$

where n is the time index. The coefficient vector **w**(n) does not equal to $\mathbf{w}_{\mathbf{opt}}$ and it fluctuates around its optimum value. If the coefficient perturbation vector is denoted by, **v**(n), then the actual coefficient vector w(n), may be modeled as follows,

$$\mathbf{w}(n) = \mathbf{w}_{opt} + \mathbf{v}(n) \tag{3}$$

The coefficient vector in equation (2) satisfies the following time recursion updates,

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \hat{\mathbf{R}}(n)^{-1}\,\mathbf{y}(n)\,e(n) \tag{4}$$

$$e(n) = x(n) - \mathbf{w}(n-1)^{T}\,\mathbf{y}(n) \tag{5}$$

$$e_{opt} = x(n) - \mathbf{w}^{T}_{opt}\,\mathbf{y}(n) \tag{6}$$

where e(n) is the equalizer error, e_{opt} is the equalizer optimal error, and $\hat{R}(n)$ and $\hat{q}(n)$ are the channel output ACM and the CCV of the channel output y(n), and the desired sequence x(n), respectively. The ACM, $\hat{R}(n)$, can be computed according to the following time recursion,

$$\hat{\mathbf{R}}(n) = \lambda\,\hat{\mathbf{R}}(n-1) + \mathbf{y}(n)^{T}\,\mathbf{y}(n) \tag{7}$$

II- EQUALIZERS INITIALIZATION

The coefficient perturbation vector v(n), defined in equation (3) can be analytically obtained by premultiplying (4) by $\hat{R}(n)$ to obtain,

$$\hat{\mathbf{R}}(n)\;\mathbf{w}(n) = \hat{\mathbf{R}}(n)\,\mathbf{w}(n-1) + \mathbf{y}(n)\,e(n) \tag{8}$$

Substituting (7) for $\hat{R}(n)$ in the R.H.S. of (8) and eq. (4) for e(n), we obtain

$$\hat{R}(n)\ w(n) = \lambda\ \hat{R}(n-1)\ w(n-1) + y(n)\,x(n) \tag{9}$$

Similar expression can be obtained for $\hat{R}(n)w_{opt}$ by employing eq.(1), and the quantity $\hat{R}(n)v(n)$ is obtained from eq. (3) by subtracting $\hat{R}(n)w_{opt}$ from $\hat{R}(n)w(n)$ given by (9). Premultiplying the result by $\hat{R}(n)^{-1}$ yields a difference equation for v(n), closed form solution of which is,

$$v(n) = \lambda^{n}\ \hat{R}(n)^{-1}\ \hat{R}(0)\ v(0) + \hat{R}(n)^{-1} \sum_{k=1}^{n} \lambda^{n-k}\ y(n)\ e_{opt} \tag{10}$$

The first term in (10), which is dominant during the start-up period, depends on the choice of $\hat{R}(0)$. It is selected as a scalar matrix having a small δ as its diagonal element. Too small a value however, slows down the convergence rate, due to finite word effects. Figure 1 shows average MSE versus n for two different values of δ. Further reduction of δ was shown to have negligible effect on the the average MSE. As shown in [4] both of these curves are inferior to the ORAL equalizer performance.

III- THE MEAN SQUARE ERROR

Recall [4] the total steady -state MSE e_{tot}, of the adaptive algorithm is greater than e_{opt}, by a value of the misadjustment MSE

$$e_{tot} = e_{opt} + \left(\frac{1-\lambda}{1+\lambda}P\right)\ e_{opt} \tag{11}$$

where λ is the forgetting factor and P is the equalizer order.

The following table shows the simulation results of the total output error MSE for both MTO and ORAL equalizers, and the theoretical values, e_{th}, computed from eq.(11). It is clear that the ORAL algorithm is closer to the terotical results than MTO.

λ	MTO	ORAL	e_{th}
0.99	1.42	1.24	1.04
0.98	2.85	1.12	1.09
0.97	4.34	1.10	1.14
0.96	5.88	1.13	1.18
0.95	7.49	1.22	1.23
0.94	9.18	1.37	1.28
0.93	10.95	1.62	1.33
0.92	12.80	1.99	1.38
0.91	14.74	2.54	1.42
0.90	16.78	3.27	1.47

The error values are scaled by 1

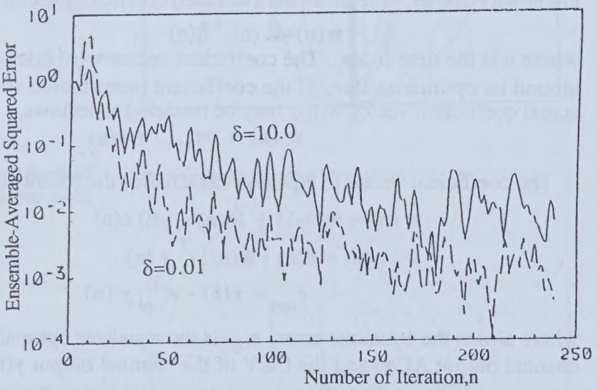


Figure 1 The effect of different valuesof δ on the convergence of the MTO equalizer, with λ=1.0 and P=11.

REFERENCES

[1] E. H. Satorius and J. Pack, "Application of Least Squares Lattice Algorithm to Adaptive Equalization," IEEE Trans. Commun., vol. COM-29, pp. 136-142, Feb. 1981.

[2] B. M. Sallam, S. B. Kesler, "Comparison of Two Modified RLS Lattice Equalizers Under Nonstationary Environments", the 31-Midwest Symposium on Circuits and Systems, St. Louis, MO, Aug. 1988.

[3] P. Strobach, "Pure Order Recursive Least Squares Ladder Algorithms," IEEE Trans. on Acoustics, Speech, and Signal Processing, vol. ASSP-34 Aug. 1986.

[4] E. Eleftheriou and D. D. Falconer, "Tracking Properties and Steady State Performance of RLS Adaptive Filter Algorithms," IEEE Transaction on Acoustics, Speech, and Signal Processing, vol. ASSP-34 Oct. 1986.

ADAPTIVE AVERAGE FILTERS FOR IMAGE PROCESSING

BO CHANG AND MORTON KANEFSKY
Department of Electrical Engineering
University of Pittsburgh
Pittsburgh, PA 15261

Many images contain sharp object edges and narrow strips of different orientations which can represent important information and should be preserved during the filtering phase. Chen and Yeh [1] tested seven filtering algorithms with respect to their noise removing and edge-preserving ability and found that the best all around results were obtained with the median and the KAVE filters [2]. Nieminen et al. suggested a class of detail preserving filters called FMH filters which are combinations of median filters [3]. In this paper, we propose a filter which will preserve details better than these filter's while efficiently removing noise. This filter uses feature detectors to adaptively switch between average filters. The image is thresholded into Q-1 (Q is the number of intensity levels) binary images, each of which is filtered separately with their outputs added.

The goal is to design a filtering algorithm which behaves as a linear filter for noise reduction within an object or background, as a KAVE filter to the background near boundaries and which preserves object boundaries. This method called an adaptive average filter (AAF) is pictured in Figure (1). The detector differentiates four kinds of pixels. "1"-feature pixels, "0"-feature pixels, non-feature pixels, and blank-space pixels. If it is part of a "1"-feature or blank-space, a pixel is left unprocessed. If it is a member of "0"-feature, it is KAVE filtered. A pixel is average filtered only if it is a non-feature pixel, i.e. part of a homogeneous region. We adopt the convention that in a binary image, an object is denoted by "1"s and background is denoted by "0"s.

A line in a binary image is defined as any collection of "1"s where each member is "eight connected" to at least one other member "1". We define the distance between pixels with coordinates (m,n) and (l,k) as $\mathrm{MAX}\{ m-l , n-k \}+1$. The length of a line is defined as the maximum distance between any two members in that line. A pixel is defined as a member of a "1"-feature if it can be regarded as the end pixel in a line of length M. A "1" pixel can be either a member of a "1"-feature or a non-feature. A "1"-feature member, is most likely part of an object and should be retained as is. Non-feature "1"s are average filtered.

A blank-space pixel is a "0" pixel totally surrounded by "0"s in a square window. Such pixels are retained unchanged. A "0"-feature is introduced in order to identify background boundaries so that they will be KAVE filtered. If the window contains some "1" pixels but the "0" pixel at the center belongs to one of a set of cluster patterns, then it is identified as a "0"-feature. Therefore the definition can be picture dependent.

A "0" pixel has three possibilities. The detector first determines whether or not it is a blank-space pixel. If not, then the detector determines whether or not it is a member of a "0"-feature. If it is, then this "0" is likely a boundary pixel and therefore it is KAVE filtered. Otherwise it is a non-feature pixel and is average filtered. This procedure is indicated by the detector function in Figure (1). Note that in this work, a "1"-feature is identified by determining connectivity to the boundary of an NxN window, where N=2M-1.

By defining different "1"-features and "0"-features, various AAF's can be formed. For example AAF8 refers to an AAF in which a line of length no less than 8 is considered as a "1"-feature, i.e. we use a 9x9 window for detection. As to the definition of a "0"-feature, we have more options. For example, we can define a line of "0"s or a strip of "0"s as "0"-feature depending on the application. In this work, however, we define eight "0"-features which fit the shape of background boundaries in the test images used. Also by selecting different KAVE filters and average filters, different AAF's can be formed. We use a second number to indicate the window used by the average filter. Thus an AAF8-5 filter refers to an AAF filter in which the average filter uses a 5x5 window. In this work, we always used a 7x7 window KAVE filter with K equal of 20.

We used four artificial test images to study the detail preserving property, the noise removing performance, and the computation time of AAF8-5 and AAF10-5 filters. The results are compared with those of the Ms3 (a median filter with a 3x3 window), 3LH-FMH [3], and KAVE with a 7x7 window and K set to 20. The images are designed to test if the filters can preserve boundary details and strips in all directions, how this property is affected by the width of objects in the images, and how these filters respond to a region of slowly varying mean. In all cases the distortion caused by the AAF filters was less and in nearly all cases the performance in noise, as measured by the mean square error, was better.

The simulation programs written in Fortran is executed on a VAX system. The KAVE filters took about seven seconds for a 64x64 image, which is equivalent to seven minutes for a 512x512 image. The AAF filters took about two seconds to process a 64x64 binary image. If AAF filters are realized in parallel, only two minutes are required to process a 512x512 image. If they are realized in series, for a 512x512 image with 256 gray levels, it takes about 255x2 minutes, or about 9 hours.

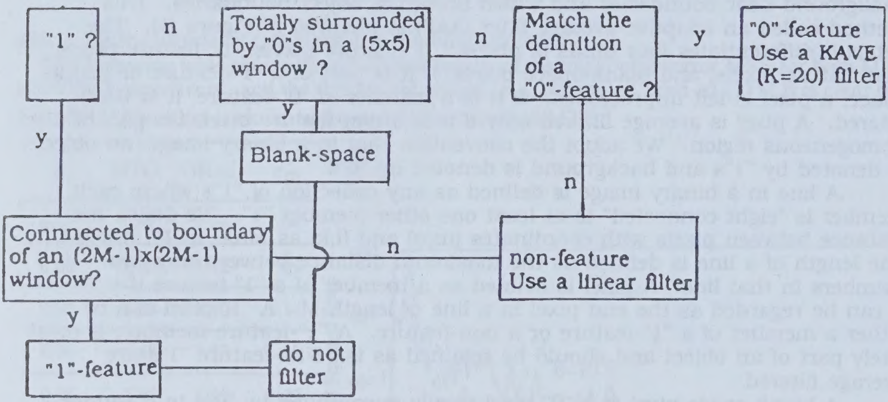

"1" ?
n
Totally surrounded by "0"s in a (5x5) window ?
n
Match the definition of a "0"-feature ?
y
"0"-feature Use a KAVE (K=20) filter
y
y
Blank-space
n
Connnected to boundary of an (2M-1)x(2M-1) window?
n
non-feature Use a linear filter
y
"1"-feature
do not filter

Figure 1. Adaptive Average Filter Structure

References

[1] R.T. Chin and C.L. Yeh, "Quantitative evaluation of some edge-preserving noise-smoothing techniques," Computer Vision, Graphics, Image Proc., Vol. 23, 1983, pp. 67-91.

[2] L.S. Davis and A. Rosenfeld, "Noise cleaning by iterated local averaging," IEEE Trans. Sys., Man, Cybern., Vol. SCM-8, 1978.

[3] A. Nieminen, P. Heinonen and Y. Neuvo, "A new class of detail-preserving filters for image processing," IEEE Trans. Patt. Anal. and Mach. Intell., Vol. PAMI-9, No. 1, Jan. 1987, pp. 74-90.

ASYMPTOTIC BOUNDS FOR FOCUSED ERROR CONTROL CODES

TOM FUJA
Department of Electrical Engineering
Systems Research Center
University of Maryland
College Park, Maryland 20742

CHRIS HEEGARD
School of Electrical Engineering
Phillips Hall
Cornell University
Ithaca, New York 14852

Abstract

Consider a channel with inputs and outputs in the field F_q ($q > 2$). We say the channel is *focused* on a set $\mathcal{B} \subset F_q^*$ if the additive noise generated by the channel is likely to lie in $\mathcal{B}$. Our concern is the construction of codes that are appropriate for such channels. We say a code is (t_1, t_2)-focused on $\mathcal{B}$ if it can correct up to $t_1 + t_2$ errors provided at most t_1 of those errors lie outside of $\mathcal{B}$; the strategy is to offer different levels of protection against "common" and "uncommon" errors and so provide novel tradeoffs between performance and rate. In this paper we are primarily concerned with computing bounds on the rates of focused codes with very large blocklengths.

I. Introduction and Motivation

When a symbol from a codeword over F_q is transmitted over a noisy channel, there are $q-1$ different noise symbols that can afflict the transmitted field element. "Traditional" error control codes – designed with respect to the Hamming metric – treat each of these $q-1$ possibilities the same – as simplying representing a generic "error."

In many data communication and data storage systems there are some errors that are much more common than others. Consider, for instance, a random access memory system that employs "byte-wide" RAM chips; that is, each chip puts out $b \geq 2$ bits at a time. It has been suggested that codes over F_{2^b} be used for such systems, with each chip's output constituting one element of F_{2^b}. Now the vast majority of chip failures are single-cell failures and so would affect only one bit per byte; thus, most of the symbol errors that would confront such a code would be one of the b field elements with exactly one non-zero bit in its binary representation. Of course, there are failure mechanism that can cause multiple bit failures – catastrophic whole-chip failures, for instance – so we would need to provide *some* protection against arbitrary errors; however, providing the same degree of protection against common and uncommon errors is not efficient.

In this and in many other applications it is desirable to "focus" the capabilities of a code on a class of particularly common errors. In a previous paper [4] we showed techniques for constructing such codes and gave preliminary results on rate bounds. In this paper we look more closely at the fundamental problem from an information theoretic point of view and use the insight gained to compute bounds on the rates of "focused codes" for asymptotically large blocklength.

II. Shannon Theory for Focused Codes

Consider the following channel model for storage or transmission. A character $x \in F_q$ is to be transmitted and the character $Y = x + Z \in F_q$ is received. It is assumed that the random error, $Z \in F_q$, is independent of the input, X, and the probability of error, $\Pr(Z \neq 0) = \epsilon$. In a typical model for a symmetric channel, the conditional error probability, $\Pr(Z = z | Z \neq 0) = 1/(q-1)$, is uniform over the error values where $z \in F_q^* = F_q - \{0\}$. We say that a channel is *focused on a subset* $\mathcal{B} \subset F_q^*$ if $\Pr(Z \in \mathcal{B} | Z \neq 0) > |\mathcal{B}|/(q-1)$. In particular, we consider the following distribution for the focused error Z

$$\Pr(Z = z) = \begin{cases} 1-\epsilon, & \text{if } z = 0, \\ \epsilon(1-\gamma)/|\mathcal{B}|, & \text{if } z \in \mathcal{B}, \\ \epsilon\gamma/|\mathcal{B}^c|, & \text{if } z \in \mathcal{B}^c, \end{cases}$$

where $\mathcal{B}^c \triangleq F_q^* - \mathcal{B}$.

We call such an error model the *symmetric focused channel* (SFC) for the focus set $\mathcal{B} \subset F_q^*$. Note that the cases of interest have parameters $0 \leq \epsilon \leq (q-1)/q$ and $0 \leq \gamma \leq |\mathcal{B}^c|/(q-1)$. We are interested in finding the Shannon capacity of this channel as a tool in the understanding of the problem of constructing focused codes.

Theorem 1: The capacity of the SFC for the focus set $\mathcal{B} \subset F_q^*$ is given by

$$C = 1 - \left[\mathrm{h}(\epsilon) + \epsilon\left(\mathrm{h}(\gamma) + (1-\gamma)\log_q|\mathcal{B}| + \gamma\log_q|\mathcal{B}^c|\right)\right]$$

where $\mathrm{h}(x) \triangleq -x\log_q(x) - (1-x)\log_q(1-x)$ is the *binary entropy function.*

Proof: As we know [1-3], the capacity is given by

$$C = \max_{p(x)} \mathrm{I}(X;\, Y).$$

In this case, $\mathrm{I}(X;\, Y) = \mathrm{H}(Y) - \mathrm{H}(Z) \leq 1 - \mathrm{H}(Z)$ with equality if the input distribution, $p(x) = 1/q$, is uniform. Finally, to compute $\mathrm{H}(Z)$ we note that if we define

$$W \triangleq \begin{cases} 0, & \text{if } Z = 0; \\ 1, & \text{if } Z \in \mathcal{B}; \\ 2, & \text{if } Z \in \mathcal{B}^c; \end{cases}$$

then $\mathrm{H}(Z) = \mathrm{H}(Z,\, W) = \mathrm{H}(W) + \mathrm{H}(Z|W)$ and $\mathrm{H}(W) = \mathrm{h}(\epsilon) + \epsilon\mathrm{h}(\gamma)$, $\mathrm{H}(Z|W) = \epsilon(1-\gamma)\log_q|\mathcal{B}| + \epsilon\gamma\log_q|\mathcal{B}^c|$. This shows the result. QED

A related question will help in the description of asymptotic bounds on the performance of focused codes. Given bounds $0 \leq \epsilon \leq \epsilon_0$, $0 \leq \gamma \leq \gamma_0$ what is the maximum value for the entropy of the error $\mathrm{H}(Z)$? The following two lemmas provide the answer.

Lemma 2: Let $0 \leq \gamma \leq \gamma_0$ and define

$$\Gamma(\gamma) \triangleq \mathrm{h}(\gamma) + (1-\gamma)\log_q|\mathcal{B}| + \gamma\log_q|\mathcal{B}^c|$$

then $\Gamma(\gamma)$ is maximum for

$$\gamma = \min\left\{\gamma_0,\, \frac{|\mathcal{B}^c|}{(q-1)}\right\}$$

Proof: To achieve the inequality $d\Gamma/d\gamma \geq 0$ requires

$$\frac{d\Gamma}{d\gamma} = \log_q\left(\frac{1-\gamma}{\gamma}\right) - \log_q|\mathcal{B}| + \log_q|\mathcal{B}^c| \geq 0$$

or $\gamma \leq |\mathcal{B}^c|/(q-1)$. QED

Note that $\log_q|\mathcal{B}| \leq \Gamma(\gamma) \leq \log_q(q-1)$.

Lemma 3: Let $0 \leq \epsilon \leq \epsilon_0$ and define

$$\mathrm{f}(\epsilon) \triangleq \mathrm{h}(\epsilon) + \epsilon\Gamma$$

where $\log_q|\mathcal{B}| \leq \Gamma \leq \log_q(q-1)$. Then $\mathrm{f}(\epsilon)$ is maximum for

$$\epsilon = \min\left\{\epsilon_0,\, \frac{q^\Gamma}{1+q^\Gamma}\right\}$$

Proof: To achieve the inequality $df/d\epsilon \geq 0$ requires

$$\frac{df}{d\epsilon} = \log_q\left(\frac{1-\epsilon}{\epsilon}\right) + \Gamma \geq 0$$

or $\epsilon \leq q^{\Gamma}/(1+q^{\Gamma})$. QED

Corollary 4: For $0 \leq \epsilon \leq \epsilon_0$, $0 \leq \gamma \leq \gamma_0$, the entropy $\mathrm{H}(Z)$ is maximized when

$$\gamma = \min\left\{\gamma_0,\ \frac{|\mathcal{B}^c|}{(q-1)}\right\}, \quad \epsilon = \min\left\{\epsilon_0,\ \frac{q^{\Gamma(\gamma)}}{1+q^{\Gamma(\gamma)}}\right\}$$

where

$$\Gamma(\gamma) \triangleq \mathrm{h}(\gamma) + (1-\gamma)\log_q|\mathcal{B}| + \gamma\log_q|\mathcal{B}^c|.$$

Then $\mathrm{H}(Z) = \mathrm{h}(\epsilon) + \epsilon\Gamma(\gamma)$.

III. Construction of Focused Codes

In [4], we developed the notion of a *focused* error control code, appropriate for focused channels. Here, we briefly outline the pertinent results from [4].

Definition: Let $\mathcal{B} \subset F_q^*$ be a set of non-zero elements of F_q. A code that is (t_1, t_2)-focused on $\mathcal{B}$ is a code that is capable of correcting up to $t_1 + t_2$ errors provided at most t_1 of those errors lie outside $\mathcal{B}$. More precisely, such a code $\mathcal{C}$ is a set of n-tuples over F_q with the following property: There exists a function $f : F_q^n \rightarrow \mathcal{C}$ such that $f(\mathbf{c}+\mathbf{e}) = \mathbf{c}$ for any $\mathbf{c} \in \mathcal{C}$ and any $\mathbf{e} \in F_q^n$ satisfying the following two properties:
(1) $||\mathbf{e}||_{\mathrm{H}} \leq t_1 + t_2$;
(2) $||\mathbf{e}||_{\mathcal{B}^c} \leq t_1$.
(About the notation: For any q-ary n-tuple $\mathbf{x}$, $||\mathbf{x}||_{\mathrm{H}}$ is the Hamming weight of $\mathbf{x}$ – that is, the number of non-zero components of $\mathbf{x}$. By contrast, $||\mathbf{x}||_{\mathcal{B}^c}$ is the number of components of $\mathbf{x}$ that lie in $\mathcal{B}^c$.)

Note that a code that is $(0, t)$-focused on F_q^* is a "traditional" t-error correcting code.

3.1. Sufficient Conditions for Focused Codes

In this section we give sufficient conditions for a set of q-ary n-tuples to form a code that is focused on $\mathcal{B} \subset F_q^*$.

Lemma 5: Let $\mathcal{C}$ be a set of q-ary n-tuples with the following property. For any $\mathbf{c_1}, \mathbf{c_2} \in \mathcal{C}$, *at least one* of the following conditions holds:
(i.) $||\mathbf{c_1} - \mathbf{c_2}||_{\mathrm{H}} > 2t_1 + 2t_2$;
(ii.) $||\mathbf{c_1} - \mathbf{c_2}||_{\mathrm{H}} + ||\mathbf{c_1} - \mathbf{c_2}||_{\mathcal{B}^c} > 4t_1 + 2t_2$.
Then $\mathcal{C}$ is (t_1, t_2)-focused on $\mathcal{B}$.

Proof: Let $\mathcal{E}_n \triangleq \{\mathbf{x} \in F_q^n : ||\mathbf{x}||_{\mathrm{H}} \leq t_1 + t_2, ||\mathbf{x}||_{\mathcal{B}^c} \leq t_1\}$ be the set of error patterns we want to correct. Then as long as $\mathbf{c_1} + \mathbf{e_1} \neq \mathbf{c_2} + \mathbf{e_2}$ for any codewords $\mathbf{c_1}, \mathbf{c_2}$ and any $\mathbf{e_1}, \mathbf{e_2} \in \mathcal{E}_n$, the code will be (t_1, t_2)-focused on $\mathcal{B}$. Therefore, as long as all differences between codewords lie outside

$$\Delta\mathcal{E}_n \triangleq \{\mathbf{e_1} - \mathbf{e_2} : \mathbf{e_1}, \mathbf{e_2} \in \mathcal{E}_n\}$$

the code will be focused. But in fact it is easily seen that $\Delta\mathcal{E}_n \subseteq \mathcal{A}_n$, where

$$\mathcal{A}_n \triangleq \{\mathbf{x} : ||\mathbf{x}||_{\mathrm{H}} \leq 2t_1 + 2t_2, ||\mathbf{x}||_{\mathrm{H}} + ||\mathbf{x}||_{\mathcal{B}^c} \leq 4t_1 + 2t_2\}.$$

Thus the lemma is proven. QED

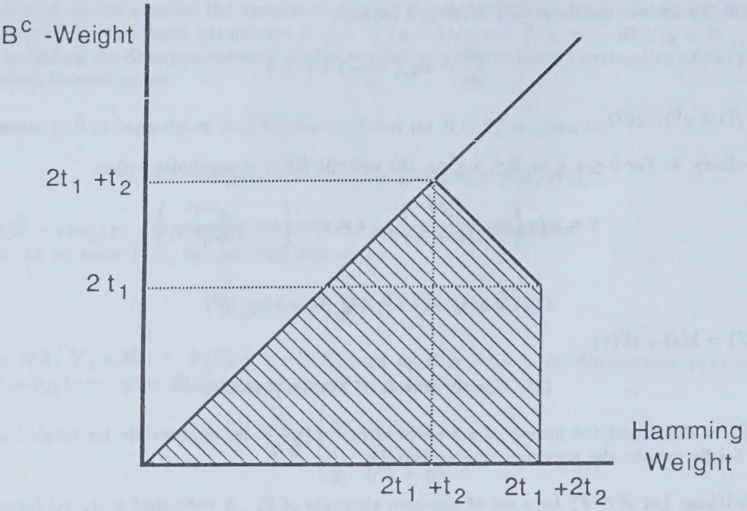


Figure 1: Graphic interpretation of Lemma 5.

The implications of this lemma are presented graphically in Figure 1. We can plot every q-ary n-tuple in two dimensions by its Hamming weight and its $\mathcal{B}^c$-weight. The lemma says that as long as no codeword difference lies in the shaded region, the code will be (t_1, t_2)-focused on $\mathcal{B}$. By comparison, to insure correction of *all* error patterns of Hamming weight $t_1 + t_2$ or less we would require that all codeword differences have Hamming weight greater than $2t_1 + 2t_2$; by lowering our requirements we've cut a "notch" in the "forbidden zone".

It's worth noting that the sufficient conditions given in Lemma 5 are not in general necessary conditions. However, they *are* necessary conditions if and only if every element of $\mathcal{B}^c$ can be expressed as the difference of two elements of $\mathcal{B}$. This means, for instance, that whenever $\mathcal{B}^c$ lies inside a proper additive subgroup of F_q, then the conditions of Lemma 5 are necessary and sufficient.

3.2. Construction of Combined Linear Focused Codes

In [4] we demonstrated how, by concatenating three different codes over three (possibly different) fields in a special way, we can construct a code that is (t_1, t_2)-focused on a set $\mathcal{B}$ for many different choices of $\mathcal{B}$. We call such codes ***combined linear focused codes***. In this section we briefly review that procedure.

Suppose we wish to construct a code with blocklength n over F_{2^b} that is (t_1, t_2)-focused on the set of odd-weight symbols; that is, the focus set $\mathcal{B}$ consists of all the elements of F_{2^b} with a binary representation containing an odd number of 1's. (Note that this would include the set of single-bit errors.) We proceed as follows. Let $\mathcal{C}_1$ be a binary code of blocklength n with minimum distance $d_1 = 2t_1 + 2t_2 + 1$; let $\mathcal{C}_2$ be a code over $F_{2^{b-1}}$ with blocklength n and minimum distance $d_2 = 2t_1 + t_2 + 1$. Then we construct a codeword in our focused code by first taking a codeword from $\mathcal{C}_2$ and adding one bit to each code symbol such that the resulting n-tuple over F_{2^b} has the following property: If you take the parity of each symbol – and so arrive at a binary n-tuple – then that binary n-tuple lies in $\mathcal{C}_1$.

We prove that the code so constructed is (t_1, t_2)-focused by describing the following decoding algorithm. Given a received 2^b-ary n-tuple, compute the binary n-tuple consisting of the mod-two sums of each of the received symbols. Map this binary n-tuple onto the closest codeword in $\mathcal{C}_1$. Wherever the binary n-tuple differs from the closest codeword in $\mathcal{C}_1$, mark the corresponding symbol as an erasure. Then, strip off the last bit from each received symbol and pass the resulting n-tuple over $F_{2^{b-1}}$ (along with the erasure locations) to a decoder for $\mathcal{C}_2$.

This algorithm will correct any combination of $t_1 + t_2$ errors provided no more than t_1 of those errors have even weight (and so must be corrected without benefit of erasure).

This technique easily generalizes to codes over F_{p^b}. We first construct a code over F_p with blocklength b that is capable of *detecting* any error in the focus set; call this code $\mathcal{C}_0$, and assume it is a $(b, k_0 = b - r_0)$ code. (In our example above, $\mathcal{C}_0$ is just a $(b, b-1)$ parity check code.) Then $\mathcal{C}_1$ is a code over $F_{p^{r_0}}$ with blocklength n and minimum distance $d_1 = 2t_1 + 2t_2 + 1$, and $\mathcal{C}_2$ is a code over $F_{p^{k_0}}$ with blocklength n and minimum distance $d_2 = 2t_1 + t_2 + 1$. The concatenation of the codes is similar to the example above and it can be shown that the rate R of the resulting focused code is given by

$$R = (1 - R_0)R_1 + R_0 R_2.$$

IV. Asymptotic Bounds for Focused Codes

In this section we compute bounds on the rates of focused codes for asymptotically large blocklengths. In the first part of this section we compute bounds that are applicable to *any* focused code. In the second part we compute asymptotic bounds for codes constructed using the technique outlined in Section 3.2.

4.1. Bounds for Arbitrary Focused Codes

The *rate* R of a code $\mathcal{C}$ over F_q is defined as $R \triangleq n^{-1}\log_q|\mathcal{C}|$. For a given blocklength n, non-negative integers t_1 and t_2, field F_q, and focus set $\mathcal{B}$, we will let $R_n^*(t_1, t_2)$ denote the highest rate of any code of blocklength n over F_q that is (t_1, t_2)-focused on $\mathcal{B}$ – i.e.,

$$R_n^*(t_1, t_2) \triangleq \sup\{R : \text{There exists a code of rate } R \text{ and blocklength } n \text{ that is } (t_1, t_2) - \text{focused on } \mathcal{B}\}.$$

In [4], it was shown that $R_n^*(t_1, t_2)$ can be bounded as follows:

$$1 - \frac{1}{n}\log_q|\mathcal{A}_n| \le R_n^*(t_1,\ t_2) \le 1 - \frac{1}{n}\log_q|\mathcal{E}_n|.$$

Here, $\mathcal{A}_n$ and $\mathcal{E}_n$ are as defined in Section 3.1 –

$$\mathcal{A}_n \triangleq \{\mathbf{x} : ||\mathbf{x}||_{\mathrm{H}} \le 2t_1 + 2t_2,\ ||\mathbf{x}||_{\mathrm{H}} + ||\mathbf{x}||_{\mathcal{B}^c} \le 4t_1 + 2t_2\}$$

and

$$\mathcal{E}_n \triangleq \{\mathbf{x} : ||\mathbf{x}||_{\mathrm{H}} \le t_1 + t_2,\ ||\mathbf{x}||_{\mathcal{B}^c} \le t_1\}.$$

Note that the upper bound is of the form of a "sphere–packing" or "Hamming" bound and the lower bound is in the form of a "Gilbert-Varshamov" bound [5]. These bounds can be combined with the results of the entropy maximization question considered in the last section to obtain bounds on the performance of focused codes for large n. Specifically, define

$$R^*(\alpha,\ \beta) \triangleq \limsup_{n\to\infty} R_n^*(\alpha n,\ \beta n).$$

Then the following two theorems provide "focused" versions of the asymptotic Gilbert-Varshamov and Hamming bounds.

Theorem 6: $R^*(\alpha,\ \beta) \ge 1 - \mathrm{h}(\epsilon) - \epsilon\Gamma(\gamma)$ where

$$\Gamma(\gamma) \triangleq \mathrm{h}(\gamma) + (1-\gamma)\log_q|\mathcal{B}| + \gamma\log_q|\mathcal{B}^c|$$

and

$$\epsilon = \min\{2\alpha + 2\beta, \frac{q^{\Gamma(\gamma)}}{1+q^{\Gamma(\gamma)}}\}, \quad \gamma = \min\{\frac{4\alpha + 2\beta - \epsilon}{\epsilon}, \frac{|\mathcal{B}^c|}{(q-1)}\}.$$

Proof:

$$\begin{aligned}
|\mathcal{A}_n| &= \sum_{i=0}^{2(\alpha+\beta)n} \sum_{j=0}^{\min(i,(4\alpha+2\beta-i)n)} \binom{n}{i}\binom{i}{j}(|\mathcal{B}^c|)^j|\mathcal{B}|^{i-j} \\
&\leq K \max\left\{\binom{n}{i}\binom{i}{j}(|\mathcal{B}^c|)^j|\mathcal{B}|^{i-j} : 0 \leq i \leq 2(\alpha+\beta)n, 0 \leq j \leq \min(i,(4\alpha+2\beta-i)n)\right\} \\
&\leq K \max\left\{q^{nE(i,j)} : 0 \leq i \leq 2(\alpha+\beta)n, 0 \leq j \leq \min(i,(4\alpha+2\beta-i)n)\right\},
\end{aligned}$$

where

$$E(i,j) = h(\frac{i}{n}) + \frac{i}{n}h(\frac{j}{i}) + \frac{j}{n}\log_q|\mathcal{B}^c| + \frac{i-j}{n}\log_q|\mathcal{B}|$$

and K is equal to the number of terms in the double sum. Note that to achieve the last inequality we've made use of the well-known

$$\binom{n}{i} \leq q^{n[h(\frac{i}{n})]}.$$

Note also that K is a polynomial in n and so

$$\lim_{n\to\infty} \frac{1}{n}\log_q(K) = 0.$$

The maximum is found by maximizing the exponent

$$h(\epsilon) + \epsilon h(\gamma) + \epsilon\gamma\log_q|\mathcal{B}^c| + \epsilon(1-\gamma)\log_q|\mathcal{B}|$$

where $\epsilon = i/n$ and $\gamma = j/i$. The values of ϵ and γ that maximize the exponent are given in the theorem and follow from Lemmas 2 & 3. QED

Theorem 7: $R^*(\alpha,\ \beta) \leq 1 - \mathrm{h}(\epsilon) - \epsilon\Gamma(\gamma)$, where

$$\Gamma(\gamma) \triangleq \mathrm{h}(\gamma) + (1-\gamma)\log_q|\mathcal{B}| + \gamma\log_q|\mathcal{B}^c|,$$

and

$$\epsilon = \min\left\{\alpha + \beta,\ \frac{q^{\Gamma(\gamma)}}{1+q^{\Gamma(\gamma)}}\right\}, \quad \gamma = \min\left\{\frac{\alpha}{\epsilon},\ \frac{|\mathcal{B}^c|}{(q-1)}\right\}.$$

Proof:

$$\begin{aligned}
|\mathcal{E}_n| &= \sum_{i=0}^{(\alpha+\beta)n} \sum_{j=0}^{\min(i,\alpha n)} \binom{n}{i}\binom{i}{j}(|\mathcal{B}^c|)^j|\mathcal{B}|^{i-j} \\
&\geq \max\left\{\binom{n}{i}\binom{i}{j}(|\mathcal{B}^c|)^j|\mathcal{B}|^{i-j} : 0 \leq i \leq (\alpha+\beta)n, 0 \leq j \leq \min(i,\alpha n)\right\} \\
&\geq \max\left\{q^{nE(i,j)} : 0 \leq i \leq (\alpha+\beta)n, 0 \leq j \leq \min(i,\alpha n)\right\}
\end{aligned}$$

where

$$E(i,j) \triangleq h(\frac{i}{n}) + \frac{i}{n}h(\frac{j}{i}) + \frac{j}{n}\log_q|\mathcal{B}^c| + \frac{i-j}{n}\log_q|\mathcal{B}| + o_n$$

and o_n goes to 0 as $n \to \infty$. The last inequality follows from the bound

$$\binom{n}{i} \geq q^{n[h(\frac{i}{n})+o_n]}.$$

The maximum is found by maximizing the exponent

$$h(\epsilon) + \epsilon h(\gamma) + \epsilon\gamma \log_q |\mathcal{B}^c| + \epsilon(1-\gamma)\log_q |\mathcal{B}|]$$

where $\epsilon = i/n$, $\gamma = j/i$. The values of ϵ and γ that maximize the exponent are given in the theorem and again follow from Lemmas 2 & 3. QED

4.2. Bounds for Combined Linear Focused Codes

In Section 3.2. we described a technique for constructing focused codes with blocklength n over F_{p^b} by combining three different codes in a specific way. In this section we compute bounds on the rates that can be achieved using this technique for asymptotically large n.

Using this technique, we note that the choice of the focus set $\mathcal{B}$ determines the code $\mathcal{C}_0$ as well as the fields over which $\mathcal{C}_1$ and $\mathcal{C}_2$ are formed; $\mathcal{C}_0$ is a $(b, k_0 = b - r_0)$ code over F_p capable of detecting any error in $\mathcal{B}$, $\mathcal{C}_1$ is a code over $F_{q_1} (q_1 = p^{r_0})$ with blocklength n and minimum distance $d_1 = 2t_1 + 2t_2$, and $\mathcal{C}_2$ is a code over $F_{q_2} (q_2 = p^{k_0})$ with blocklength n and minimum distance $d_1 = 2t_1 + t_2$. For a given choice of $\mathcal{B}$ (and therefore for a given choice of $\mathcal{C}_0$, which we assume has rate R_0) we can apply the traditional asymptotic rate bounds to R_1 and R_2 to derive asymptotic bounds for combined linear focused codes. Some of the bounds thus derived are given in Theorem 8.

Theorem 8: Let $\mathcal{B} \subset F_q^* \ (q = p^b)$ and let

$$R_n^c(t_1, t_2) \triangleq \sup\{R : \text{There exists a combined linear focused code of rate } R \text{ and blocklength } n \text{ that is } (t_1, t_2)-\text{focused on } \mathcal{B}\},$$

and define

$$R^c(\alpha, \beta) \triangleq \limsup_{n\to\infty} R_n^c(\alpha n, \beta n).$$

If we define the function $H(\epsilon, m)$ as

$$H(\epsilon, m) \triangleq \frac{h(\epsilon) + \epsilon \log_q(m-1)}{\log_q(m)},$$

the function $G(\epsilon, m)$ as

$$G(\epsilon, m) \triangleq \frac{\epsilon m}{m-1},$$

$\bar{R}_0 \triangleq 1 - R_0$, and $\theta_i \triangleq (q_i - 1)/q_i$ for $i = 1, 2$, then the following inequalitites hold.

Gilbert-Varshamov Bound

$$R^c(\alpha, \beta) \geq \begin{cases} 1 - \bar{R}_0 H(2\alpha + 2\beta, q_1) & \\ \qquad - R_0 H(2\alpha + \beta, q_2), & \text{if } 0 \leq 2\alpha + 2\beta \leq \theta_1 \text{ and } 0 \leq 2\alpha + \beta \leq \theta_2; \\ \bar{R}_0 - \bar{R}_0 H(2\alpha + 2\beta, q_1), & \text{if } 0 \leq 2\alpha + 2\beta \leq \theta_1 \text{ and } \theta_2 \leq 2\alpha + \beta \leq 1; \\ R_0 - R_0 H(2\alpha + \beta, q_2), & \text{if } \theta_1 \leq 2\alpha + 2\beta \leq 1 \text{ and } 0 \leq 2\alpha + \beta \leq \theta_2; \\ 0, & \text{if } \theta_1 \leq 2\alpha + 2\beta \leq 1 \text{ and } \theta_2 \leq 2\alpha + \beta \leq 1. \end{cases}$$

Hamming Bound

$$R^c(\alpha, \beta) \leq \begin{cases} 1 - \bar{R}_0 H(\alpha + \beta, q_1) & \\ \qquad - R_0 H(\alpha + \tfrac{1}{2}\beta, q_2), & \text{if } 0 \leq \alpha + \beta \leq \theta_1 \text{ and } 0 \leq \alpha + \tfrac{1}{2}\beta \leq \theta_2; \\ \bar{R}_0 - \bar{R}_0 H(\alpha + \beta, q_1), & \text{if } 0 \leq \alpha + \beta \leq \theta_1 \text{ and } \theta_2 \leq \alpha + \tfrac{1}{2}\beta \leq 1; \\ R_0 - R_0 H(\alpha + \tfrac{1}{2}\beta, q_2), & \text{if } \theta_1 \leq \alpha + \beta \leq 1 \text{ and } 0 \leq \alpha + \tfrac{1}{2}\beta \leq \theta_2; \\ 0, & \text{if } \theta_1 \leq \alpha + \beta \leq 1 \text{ and } \theta_2 \leq \alpha + \tfrac{1}{2}\beta \leq 1. \end{cases}$$

Plotkin Bound

$$R^c(\alpha,\beta) \le \begin{cases} 1-\bar{R}_0 G(2\alpha+2\beta,q_1) & \\ \quad -R_0 G(2\alpha+\beta,q_2), & \text{if } 0 \le 2\alpha+2\beta \le \theta_1 \text{ and } 0 \le 2\alpha+\beta \le \theta_2; \\ \bar{R}_0 - \bar{R}_0 G(2\alpha+2\beta,q_1), & \text{if } 0 \le 2\alpha+2\beta \le \theta_1 \text{ and } \theta_2 \le 2\alpha+\beta \le 1; \\ R_0 - R_0 G(2\alpha+\beta,q_2), & \text{if } \theta_1 \le 2\alpha+2\beta \le 1 \text{ and } 0 \le 2\alpha+\beta \le \theta_2; \\ 0, & \text{if } \theta_1 \le 2\alpha+2\beta \le 1 \text{ and } \theta_2 \le 2\alpha+\beta \le 1. \end{cases}$$

Numerical Results

On the next two pages are some graphs of the bounds derived. It is interesting to note that, asymptotically, very little is given up in terms of rate in providing "focused" protection on the odd-weight errors as compared to the single-bit errors

§REFERENCES

[1] Robert Gallager, *Information Theory and Reliable Communications*, John Wiley & Sons, 1968.

[2] Imre Csiszár and János Körner, *Information Theory: Coding Theorems for Discrete Memoryless Channels*, Academic Press, 1981.

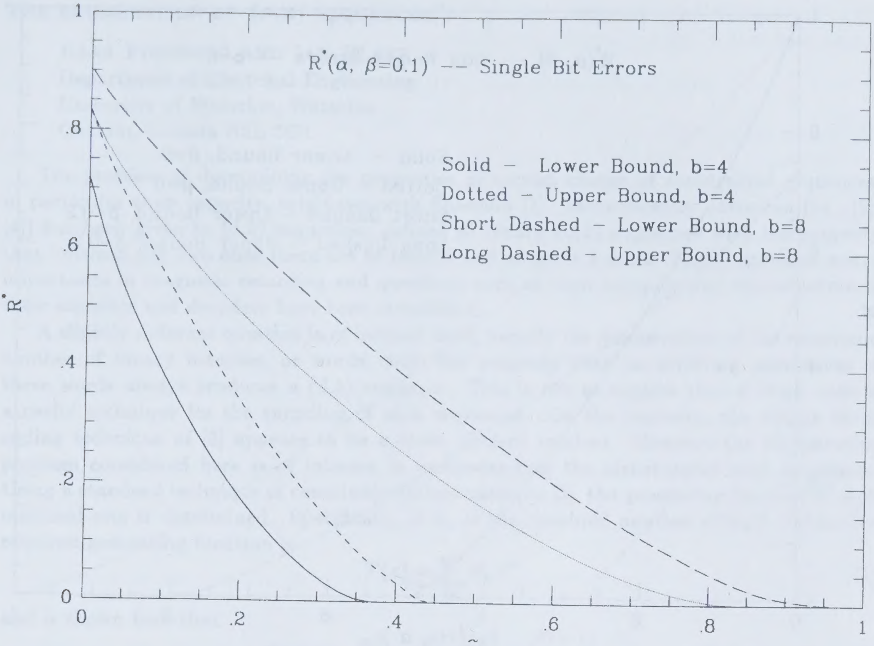
[3] Richard Blahut, "Principles and Practice of Information Theory," Addison-Wesley, 1987.

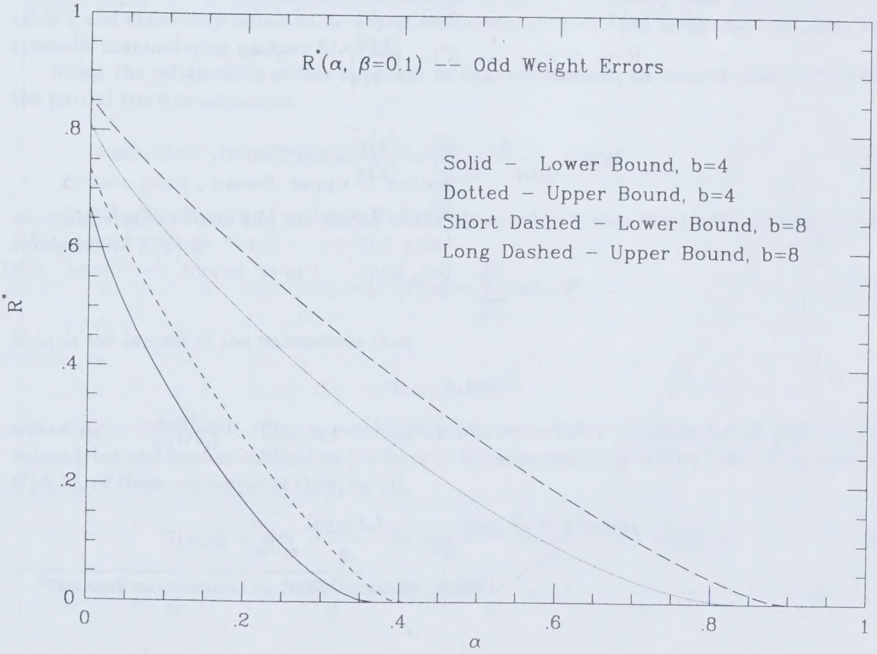
[4] Tom Fuja and Chris Heegard, "Focused Error Control Codes," 1988 Conference on Information Sciences and Systems, Princeton University, March 16–18, 1988.

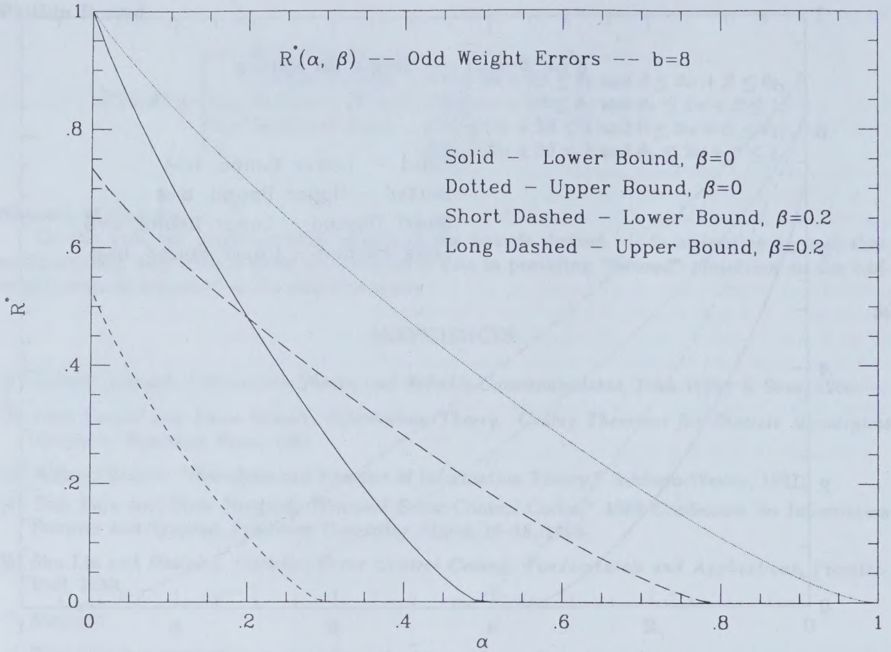
[5] Shu Lin and Daniel J. Costello, *Error Control Coding: Fundamentals and Applications*, Prentice-Hall, 1983.

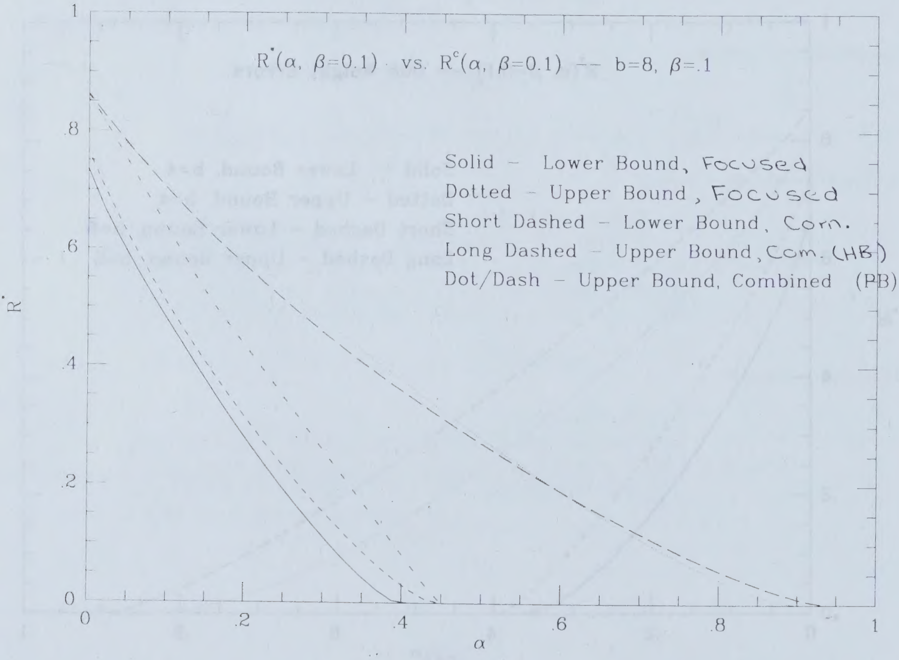
Support

Tom Fuja is supported by a grant from the Minta Martin Foundation.
Chris Heegard is supported by NSF grants ECS83-52220 and MIP87-11793.


$R^*(\alpha, \beta=0.1)$ -- Single Bit Errors
Solid - Lower Bound, b=4
Dotted - Upper Bound, b=4
Short Dashed - Lower Bound, b=8
Long Dashed - Upper Bound, b=8
R^*
1
.8
.6
.4
.2
0
0
.2
.4
.6
.8
1


$R^*(\alpha, \beta=0.1)$ -- Odd Weight Errors
Solid - Lower Bound, b=4
Dotted - Upper Bound, b=4
Short Dashed - Lower Bound, b=8
Long Dashed - Upper Bound, b=8
R^*
1
.8
.6
.4
.2
0
0
.2
.4
.6
.8
1
α


R*(α, β) -- Odd Weight Errors -- b=8
Solid – Lower Bound, β=0
Dotted – Upper Bound, β=0
Short Dashed – Lower Bound, β=0.2
Long Dashed – Upper Bound, β=0.2
R*
α
0
.2
.4
.6
.8
1


R*(α, β=0.1) vs. R^c(α, β=0.1) -- b=8, β=.1
Solid – Lower Bound, Focused
Dotted – Upper Bound, Focused
Short Dashed – Lower Bound, Com.
Long Dashed – Upper Bound, Com. (HB)
Dot/Dash – Upper Bound, Combined (PB)
R*
α
0
.2
.4
.6
.8
1

THE ENUMERATION OF (D,K) SEQUENCES.[1]

KLAS FORSBERG AND IAN BLAKE
Department of Electrical Engineering
University of Waterloo, Waterloo
Ontario, Canada N2L 3G1

The problem of determining the properties of certain classes of constrained sequences, in particular their capacity, originates with Shannon [1]. More recently attention (eg. [2]-[4]) has been given to (d,k) sequences, defined as binary {0,1} sequences with the property that between any two ones there are at least d and at most k zeros. Such sequences are of importance in magnetic recording and questions such as their capacity and the structure of their encoders and decoders have been considered.

A slightly different question is of interest here, namely the enumeration of the maximum number of binary n-tuples, or words, with the property that an arbitrary catenation of these words always produces a (d,k) sequence. This is not to suggest that a block code is a useful technique for the encoding of such sequences. On the contrary, the sliding block coding technique of [3] appears to be a more efficient method. However the enumeration problem considered here is of interest in understanding the structure of such sequences. Using a standard technique of combinatorial enumeration [5], the generating function of such maximal sets is determined. Specifically, if d_n is the maximal number of such words, the required generating function is

$$T(x) = \sum_n d_n x^n \tag{1}$$

and is shown here that

$$T(x) = \frac{x(1-x^{i+1})(x^d - x^{k+1-i})}{(1-x)(1-x-x^{d+1}+x^{k+2})} \tag{2}$$

where $i = \lfloor \frac{k-d}{2} \rfloor$. As an example, enumeration of d_n for the widely used (2,7) code are in table 1 and these were obtained by Taylor series expansion of $T(x)$ in equation (2) using the symbolic manipulation package MAPLE.

To see the relationship of this approach to that of Shannon, we observe that if $T(x)$ has the partial fraction expansion

$$T(x) = \frac{q(x)}{p(x)} = \sum_{i=1}^{m} \frac{A_i}{(1-\alpha_i x)} + h(x) \tag{3}$$

assuming distinct roots and the degree of $p(x)$ is $m = k+3$, then the coefficient of x_n in the expansion of $T(x)$ is

$$d_n = [x^n]T(x) = \sum_{i=1}^{m} A_i(\alpha_i)^n. \tag{4}$$

If α_1 is the largest of the parameters then

$$d_n \sim A_1(\alpha_1)^n \tag{5}$$

where $A_1 = \frac{-\alpha_1 q(1/\alpha_1)}{p'(1/\alpha_1)}$. This approximation was remarkably accurate for all sets of (d,k) values tried and further analysis on the error in the approximation will be done. The capacity, $C(d,k)$, of these sequences is then, by [1],

$$C(d,k) = \lim_{n\to\infty} \frac{\log_2(d_n)}{n} = \lim_{n\to\infty} \frac{\log_2 A_1 + n\log_2 \alpha_1}{n} = \log_2 \alpha_1. \tag{6}$$

---

[1] This work was supported by NSERC grant No. A7382.

Thus the capacity of the sequences is determined by the smallest real root of $p(x)$ or the largest real root of the reciprocal polynomial

$$p^*(x) = x^{k+3}p(1/x) = (x-1)(x^{k+2} - x^{k+1} - x^{k+1-d} + 1) \qquad (7)$$

which is a result of Shannon [1].

A question of interest is the approach of $\log_2(d_n)/n$ to capacity and the degree of accuracy of $A_1\alpha_1^n$ to d_n. Experimental results on these questions are also included in Table 1 for the (2,7) codes. For this case $\alpha_1 = 1.431343141$ and $A_1 = 0.416218199$.

Several interesting recursions for (d,k) sequences have been found. For example, it has been observed previously [4] that $C(d,\infty) = C(d-1, 2d-1)$. In a similar manner it is shown here that $C(d, 2d) = C(d+1, 3d+1)$.

n	d_n	$A_1\alpha_1^n$	$\log_2(d_n)/n$
8	7	7.3	0.3509
9	10	10.5^-	0.3691
10	15	15.0	0.3907
11	22	21.5^+	0.4054
12	31	30.8	0.4128
13	44	44.1	0.4200
14	63	63.1	0.4269
15	90	90.3	0.4328
20	542	542.2	0.4541
25	3258	3257.7	0.4668
30	19572	19571.9	0.4752
40	706434	706433.7	0.4858
50	25498172	25498171.8	0.4921

Table 1: Enumerative Properties of (2,7) Sequences ($C(2,7) = 0.5174$).

REFERENCES

[1] C. E. Shannon. "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379-423, 623-656, 1948.

[2] E. Zehavi and J. Wolf, "On Runlength Codes," *IEEE Trans. Inform. Theory*, vol. IT-34,pp. 45-54, Jan. 1988

[3] R. Adler, D. Coppersmith, and M. Hassner, "Algorithms for sliding block codes," *IEEE Trans. Inform. Theory*, vol. IT-29, pp. 5-22, Jan. 1983.

[4] J. Ashley and P. Siegel, "A Note on the Shannon Capacity of Run-Length-Limited Codes," *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 601-605, July 1987.

[5] I. P. Goulden and D. M. Jackson, "Combinatorial Enumeration," *New York: John Wiley and Sons*, 1983.

[6] C. V. Freiman and A. D. Wyner, "Optimum block codes for noiseless input restricted channels," *Inform. and Control* 7, pp. 398-415, 1964.

FOURIER TRANSFORM CODING FOR PEAK LIMITED CHANNELS

EPHRAIM FEIG
IBM Research, Yorktown Heights, New York 10598

Abstract: Fourier transform coding (FTC) is a digital implementation of frequency division multiplexing. We design FTC codes which yield equal reliability of transmission along the various frequency subchannels and satisfy a peak power constraint.

FOURIER TRANSFORM CODING

Frequency division multiplexing via Fast Fourier transformations have been proposed for increasing transmission rates in channels whose signal to noise ratios are varying functions of frequency [1 – 5]. Originally motivated by the classical water-pouring theorem [6 – 7] from information theory, they have been recently modified [8 – 10] for practical implementation. These modifications also incorporate trellis coding in the frequency domain, a method for combining two very disparate ideas for increasing transmission rates without sacrificing reliability. Such schemes are called, generically, Fourier transform coding, or FTC.

A major drawback in FTC is that average power constraints do not guarantee that the encoded signal has no spurious power bursts. Most communication channels require some peak power limitation for linear performance, and standard encoding schemes guarantee that this peak-power constraint is not violated. This is not the case with FTC. Good FTC codes must find a balance between high signal to noise (increasing power) and error rate due to peak-power violation (decreasing power). Here we design codes without sacrificing in power, but we clip the signal at a particular threshold level. The codes are designed to be tolerant to the extra error due to such clipping.

In FTC schemes, information is transmitted in blocks, each comprising of a sequence complex numbers $c_1, c_2, ..., c_N$. The channel is decomposed into N subchannels with disjoint frequency supports; the $j-$ th subchannel will transmit the information encoded in c_j. The actual encoding depends on the particular frequency response and noise characteristics of the channel, on the complexity and latency one is willing to tolerate, on the desired reliability, and other considerations. What is important for us here is that the expected value of the magnitude squared $|c_j|^2$ depends linearly on the amount of energy the code designer allows for the $j-$ th subchannel. A sequence $\hat{S}$ of length $2N$ is then formed by appending to the original sequence its reverse conjugate. The discrete Fourier transform S of $\hat{S}$ is a sequence of $2N$ real numbers. The transmitted signal is a concatenation of blocks, each of which ideally has as its sampled values the sequence S. Some modification (added redundancy) is made to account for the memory in the channel; this is a well understood technicality [2, 5, 9] which will not concern us here. The channel being peak limited, we pass the data through a limiter (clip) yielding the sequence $\tilde{S}$. This amplitude limited sequence is passed through a D/A converter, low-pass filtered, and sent to the channel. The receiver computes the inverse discrete Fourier transform of the sampled values of the various blocks, accounts for the channel frequency response, and obtains the original sequence $\hat{S}$ plus noise terms. The inverse discrete Fourier transform maps the additive white noise component to an additive white noise sequence of identical noise power spectrum, so we will ignore it in our analysis here. We will only be concerned with the error term induced by the clipping.

Figure 1 gives a simplified block diagram of the FTC scheme with clipping.

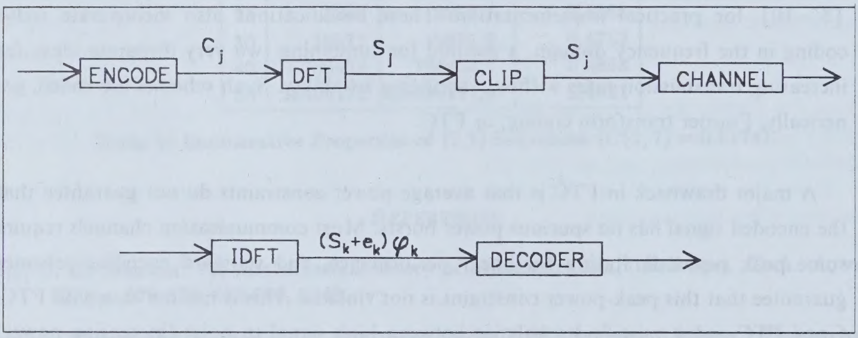


Figure 1: FTC transmitter and receiver

Data is encoded into a sequence of complex numbers c_j. These are mapped via the DFT to a sequence

$$s_k = \frac{1}{\sqrt{2N}} \left\{ \sum_{j=0}^{N-1} (c_j + c_j^*) \cos\left(\frac{2\pi jk}{2N} \right) + i \sum_{j=0}^{N-1} (c_j - c_j^*) \sin\left(\frac{2\pi jk}{2N} \right) \right\} \quad (1)$$

where c_j^* denotes the complex conjugate. The real values s_k are clipped to form

$$\tilde{s}_k = \begin{cases} s_k & | s_k | \le P \\ \pm P & | s_k | > P \end{cases} \tag{2}$$

We will write the error term due to clipping as $e_k = \tilde{s}_k - s_k$. The clipping guarantees that the channel behaves linearly. The output is sampled and inverse Fourier transformed to give the product in the frequency domain $(\hat{s}_m + \hat{e}_m)\,\Phi_m$, where Φ_m is the frequency response of the channel at frequency m and

$$\hat{e}_m = \frac{1}{\sqrt{2N}} \sum_{k=0}^{2N-1} e_k \exp\left(\frac{2\pi ikm}{2N} \right) \tag{3}$$

By construction, $\hat{s}_m = c_m$; this is the basis for the FTC scheme.

A WATERPOURING RECIPE

It was shown in [11] that the error in the decoder due to the clipping at the k – th subchannel is "approximately" normal, with mean 0 and variance

$$V_k(E, P) = \Phi_k^2 \, V(E, P) \tag{4}$$

where

$$V(E, P) = \frac{1}{E\sqrt{2\pi}} \int_P^{\infty} (x - P)^2 \exp\left(\frac{-x^2}{2E^2} \right) dx \tag{5}$$

and E^2 is the average power per frequency bin. We say "approximately" because the result is a consequence of the central limit theorem. The convergence to a normal distribution depends on the uniformity of the input power allocation. If the allocation is dominant on some narrow band of frequencies, then the tails of the distribution are inordinately heavy.

For the analysis here, it is assumed that the real and imaginary components of the constellation members are evenly spaced around 0 , and separated by distance at least σ_j. We define a separation criteria

$$\rho = \frac{\sigma_j}{\sqrt{N_j + V_j(E, P)}} \tag{6}$$

As was shown in [10], optimal transmission rate at this given reliability is given by maximizing

$$\sum_j \log_2 \left(1 + \frac{12 H_j^2 S_j^2}{\rho^2 (N_j + V_j(E, P))} \right) \tag{7}$$

Unlike the case of the optimization problem in the classical waterpouring theorem, here we impose no average power constraint, and the E in the constraint

$$\sum_j S_j^2 = E^2 \tag{8}$$

is another design parameter. As usual, we solve a variational problem by adjoining a Lagrange multiplier, and we obtain for optimal transmission

$$S_k^2 = \frac{1}{\lambda(E)} - \frac{\rho^2}{12} \frac{N_k + \Phi_k^2 V(E, P)}{|\Phi_k|^2} \tag{9}$$

whenever S_k is positive (the spectral support of the code). The pouring level λ is a function of the parameter E. With this optimal input power allocation, the transmission rate per block is

$$½ \sum_k \log_2 \frac{6}{\lambda(E) \rho^2} \frac{\Phi_k^2}{N_k + \Phi_k^2 V(E, P)} \tag{10}$$

This can be rewritten as

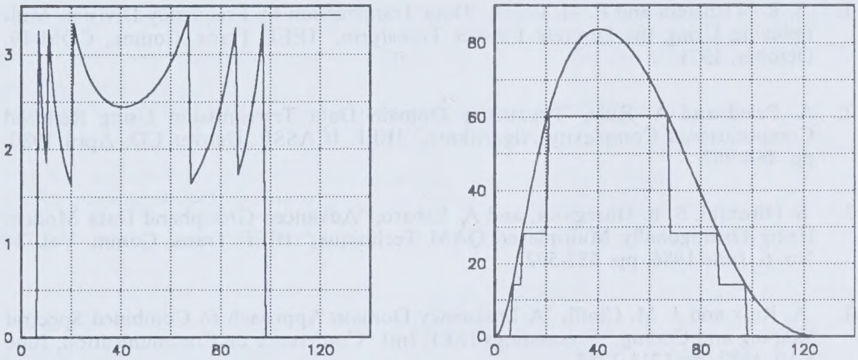
$$½ \sum_k \log_2 \left(\frac{6}{\rho^2} \right) + ½ \sum_k \log_2 \lambda(E) - ½ \sum_k \log_2 \left(\frac{\Phi_k^2}{N_k + \Phi_k^2 V(E, P)} \right) \tag{11}$$

To find the optimal total input power allocation, we differentiate with respect to E and set the result equal to 0. After some algebraic manipulation, we obtain the equation

$$\frac{B(E)}{\lambda(E)} = \sum_k \frac{\Phi_k^2}{N_k + \Phi_k^2 V(E, P)} \tag{12}$$

where $B(E)$ is the number of utilized frequency bins, which is again a function of the total input power. The functions $B(E)$ and $\lambda(E)$ are themselves very complicated functions of the Φ_k^2 and N_k.

Here we present a heuristic input power allocation strategy based on the ideas presented above. The example is taken from [10], and utilizes the method of "power pruning" introduced in [8]. In our example, the channel frequency power response is $e^{-4|w|/\pi} | \sin w|^2$, and the noise is additive white with mean squared power .005. The channel is discretized into 128 subchannels, and we impose the peak amplitude constraint $P = \pm 2.5$. We demand an error rate of 10^{-7}. Then $V(E, P) = .0012$. We obtain a code by power pruning from the constant level 3.4 and allocating 243.2 units of energy. The reason that we prune from a constant level is that the input power allocation given by equation (9) is pretty much constant on the spectral support and drops very quickly to 0. The curves of Figure 2, left, are the constant level and the the pruned allocation, as functions of frequency. The curves of Figure 2, right, are the corresponding output signal to noise power ratios.



INPUT POWER ALLOCATION

OUTPUT S/N POWER RATIOS

Figure 2: FTC for peak amplitude 2.5

In this example $V(E, P) = .0154$, and the transmission scheme utilizes 15, 29, and 47 subchannels transmitting 1, 2, and 3 bits each, respectively, for a total of 214 bits per block. The three constellations used are (*i*) the two square roots of unity, (*ii*) the four fourth roots of unity, and (*iii*) an optimally packed 8 point constellation. Their respective d_{free}'s are 2, $\sqrt{2}$, and $2 / \sqrt{4.25}$. The three levels of the output SNR for the power pruned allocation are precisely designed to achieved the prescribed reliability.

If we also use the two simple trellis codes of [12], the four state 8-PSK and eight state 16-QAM schemes, then by power pruning from the constant level 3.73, we would utilize 34 and 58 subchannels transmitting 2 and 3 bits each, respectively, for a total of 242 bits per block.

Our construction does not guarantee that all codewords are "good" (that is, are separated from other codewords by some minimum distance). We may indeed have bad codewords. But we can design our codes so that the probability of getting a bad codeword is as small as we would like it to be, while the remaining good codewords are still sufficiently far apart. Here we have come back to something reminiscent of the basic Shannon construction.

REFERENCES

1. S. B. Weinstein and P. M. Ebert, "Data Transmission by Frequency Division Multiplexing Using the Discrete Fourier Transform," IEEE Trans. Comm., COM-19, October, 1971.

2. A. Peled and A. Ruiz, "Frequency Domain Data Transmission Using Reduced Computational Complexity Algorithms," IEEE ICASSP, Denver CO, April 1980, pp. 964-967.

3. S. Hirosaki, S. B. Hasegawa, and A. Sabato, "Advanced Groupband Data Modem Using Orthogonally Multiplexed QAM Technique," IEEE Trans. Comm. Vol. 34 No. 6, June 1986, pp. 587-592.

4. A. Ruiz and J. M. Cioffi, "A Frequency Domain Approach to Combined Spectral Shaping and Coding," Proceedings IEEE Intl. Conference on Communication, June 8-10, 1987, pp.1711-1715.

5. F. Mintzer and T. Howell, "Experiments on the Use of FDDT for AC-Biase Disk Recording," IBM-RC 9429 (#41644), 6/16/82.

6. R. W. Lucky, J. Saltz, and E. J. Weldon, Principles of Data Communication, McGraw-Hill, New York, 1968, pp. 28.

7. R. Gallager, Information Theory and Reliable Communication, John Wiley and Sons, Inc., 1968, pp. 383-389.

8. E. Feig and F. Mintzer, "Sequence Transmission: Coding in the Frequency Domain," IBM-RC, 1988. Will appear in the Proceedings of the International Advanced Workshop on Sequences, Positano, Italy, June 5-11, 1988.

9. A. Ruiz, J. M. Cioffi and S. Kasturia, "Frequency Designed Coset Codes for Spectrally Constrained Channels," IBM-RC, March 10, 1988

10. E. Feig, "Coding in the Frequency Domain," in preparation.

11. E. Feig and A. Nadas, "Performance of Fourier Transform Division Multiplexing Schemes on Peak Limited Channels," IBM RC, 1988; submitted to GLOBECOM 88.

12. G. Ungerboeck, "Channel Coding with Multilevel/Phase Signals," IEEE Trans. on Information Theory, Vol. 28 No. 1, January 1982, pp. 55-67.

IMPROVING GENERALIZED MINIMUM DISTANCE DECODING BY COMPARISON WITH ERRORS-ONLY DECODING

DANA TAIPALE
University of Texas at Austin, Austin, Texas

Generalized minimum distance (GMD) decoding is a soft decision decoding algorithm that can be used on any block code for which an errors-and-erasure algorithm is known. For each transmitted codeword symbol, the GMD decoding algorithm obtains two signals from the receiver. One of these signals is an estimate of the transmitted symbol value (called a hard decision), and the other signal (typically a nonnegative scalar) indicates how reliable the hard decision is. The GMD decoding algorithm produces candidates for the decoder output. Using the hard decisions and the reliabilities, each candidate is tested to determine if the generalized distance between it and the received signals is smaller than any other codeword. A candidate with the smallest generalized distance is accepted as the decoder output. For GMD decoding, the test used to determine if the generalized distance is the smallest of all codewords is sufficient but not necessary. The test condition we present below is also sufficient (and not necessary), but it is better than the original in that it can accept an output even when the original test may fail.

Before stating the test condition, we introduce a few definitions. Denote the minimum distance of the code by d. Let α_i be the reliability of the i-th symbol in a codeword, and let S be the set of all indices i such that the i-th candidate codeword symbol does not agree with the i-th hard decision. Define e to be the cardinality of S. Now define T to be the set of indices corresponding to the d-e smallest reliabilities for codeword symbols not in S. The test condition is ,

$$\sum_{i\in S}\alpha_i < \sum_{i\in T}\alpha_i . \qquad (1)$$

The GMD decoding algorithm produces test candidates by repeated applications of an errors-and-erasures decoding algorithm. By using an errors-and-erasures decoding algorithm identical to that described by Forney, it has been shown that we will find a codeword that satisfies the condition if one exists.

To understand why (1) is still pessimistic, we must carefully examine the test condition. The test condition works by comparing the generalized distance of the candidate to the generalized distance of a competing vector. This competing vector has the smallest possible generalized distance, given that the vector disagrees with at least d symbols of the candidate. We call this a vector because it need not be a codeword. Since we are really interested in comparing the generalized distances of codewords, there is room for improvement.

To better discuss the improvement developed here, we make another definition. Let $t=\lfloor (d-1)/2 \rfloor$, where $\lfloor \bullet \rfloor$ is the greatest integer function (t is the number of errors that an errors-only decoding algorithm can correct). Also, we refer to a difference between the hard decision and a candidate codeword symbol (or a competing vector symbol) as an error. The key idea in the improved algorithm is based on the fact that there is at most one codeword with t or fewer errors (which we can find using an errors-only decoding algorithm). Let f be the number of errors in codeword found by the errors-only decoding algorithm. If the errors-only algorithm fails to find a codeword, set f equal to t. We can directly compare the generalized distance of any candidate codeword to the distance of the the codeword with f errors. Having done this, we only need to determine if the candidate has the smallest generalized distance of any codeword with at least $d\text{-}f$ errors. This will certainly be true if the candidate codeword has a smaller generalized distance than a competing vector with the smallest generalized distance, given that the vector is at (Hamming) distance d or more from the candidate *and* the vector has at least $d\text{-}f$ errors.

If the candidate codeword has t or fewer errors, then our new condition reduces to the one above. For all other candidates, however, we have a better test condition. Before writing the new test condition, we define the needed sets of indices. Define S as above. Let T' be the set of indices corresponding to the smallest $d\text{-}e + \lceil (e\text{-}f)/2 \rceil$ reliabilities not in S ($\lceil \bullet \rceil$ is the least integer function). Finally, let U be the set of indices corresponding to the smallest $\lfloor (e\text{-}f)/2 \rfloor$ reliabilities not in T (note that U may contain members of S). The candidate passes the test if,

$$\sum_{i \in S} \alpha_i < \sum_{i \in T'} \alpha_i + \sum_{i \in U} \alpha_i \,, \tag{2}$$

and if the candidate has a smaller generalized distance than the codeword with t or fewer errors.

This test condition will find acceptable codewords even when the older tests fail, hence it is not surprising to find that the original GMD errors-and erasures algorithm does not always find an acceptable codeword (if one exists). We show that, in general, the number of additional errors-and-erasures decoding attempts can quickly grow to be too large to be practical. For many channels, however, the probability of test (1) failing to correctly decode is dominated by the probability of events where $t+1$ hard decisions do not agree with the transmitted codeword. We develop an algorithm (using multiple errors-and-erasures attempts) that finds all codewords with $t+1$ or fewer errors that can satisfy (2). As a result, the probability of not decoding correctly is significantly smaller for the new algorithm (as compared to the old algorithm). In addition, we note that (2) is better than (1) even if the old GMD errors-and-erasures decoding algorithm is used.

A UNIVERSAL OUTER CODE DECODING TECHNIQUE*

J.J. METZNER
Computer Engineering Program
Department of Electrical Engineering
The Pennsylvania State University
University Park, PA 16802

ABSTRACT

Most practical decoding schemes are based on very specific linear code structures. This paper describes a linear code decoding technique of general applicability for serving as the outer code of a concatenated coder/decoder. The method is about the same order of complexity as decoding of BCH codes, is applicable to any linear code with block symbols, and can correct most patterns of d-2 or fewer erroneous inner blocks, where d is the minimum distance of the outer code. The fraction of errors in this range which it cannot correct goes down exponentially with inner block size.

I. INTRODUCTION

A recent paper has described [1] a technique for locating disagreeing pages of remotely located replicated data files. In that paper, a C-bit signature is derived from each page. Certain combinations of these signatures are then formed and communicated to the other site. The signature combination differences are used to locate the page disagreements. It was noted that the exact signature differences could be solved for after the disagreements were located, but this was not of much interest for the application. In this paper we show how a modification of the techniques in [1] can be applied to design an error correcting decoder.

A symbol of the outer code of a concatenated code [2] is a block B_i of digits from a smaller symbol alphabet. These play a role analogous to the signature in [1], except in this case they correspond uniquely to the underlying data. The decoding technique requires that the nonzero error patterns (analogous to signature differences in [1]) for the different B_i blocks

---

*This research has been supported by the National Science Foundation under grant 8796229.

be linearly independent vectors. Independence of error patterns is not normally a satisfactory assumption. For example, two random single errors in different C-bit blocks have probability 1/C of being identical. When the blocks themselves are inner codes, the chance of error independence is somewhat greater for common error patterns because patterns of small numbers of errors are corrected, and the erroneous blocks are left with a significant number of errors. The chance of block error independence can be greatly enhanced by pseudo-randomly transforming the inner blocks before transmission, with different transformations for different blocks, then inverse transforming at the receiver after inner decoding but prior to outer decoding. In this way small numbers of residual errors after inner decoding are transformed into almost random, statistically independent patterns for the outer decoder. If the resulting error patterns were truly random and statistically independent, the probability of linear dependence of t block error patterns can be shown [1] to be less than $2^{-(C-t)}$ for binary codes, where C is the number of bits per block.

II. DEFINITIONS

n_0 = number of outer code blocks.

k_0 = number of outer code data blocks.

k_1 = number of data bits per block.

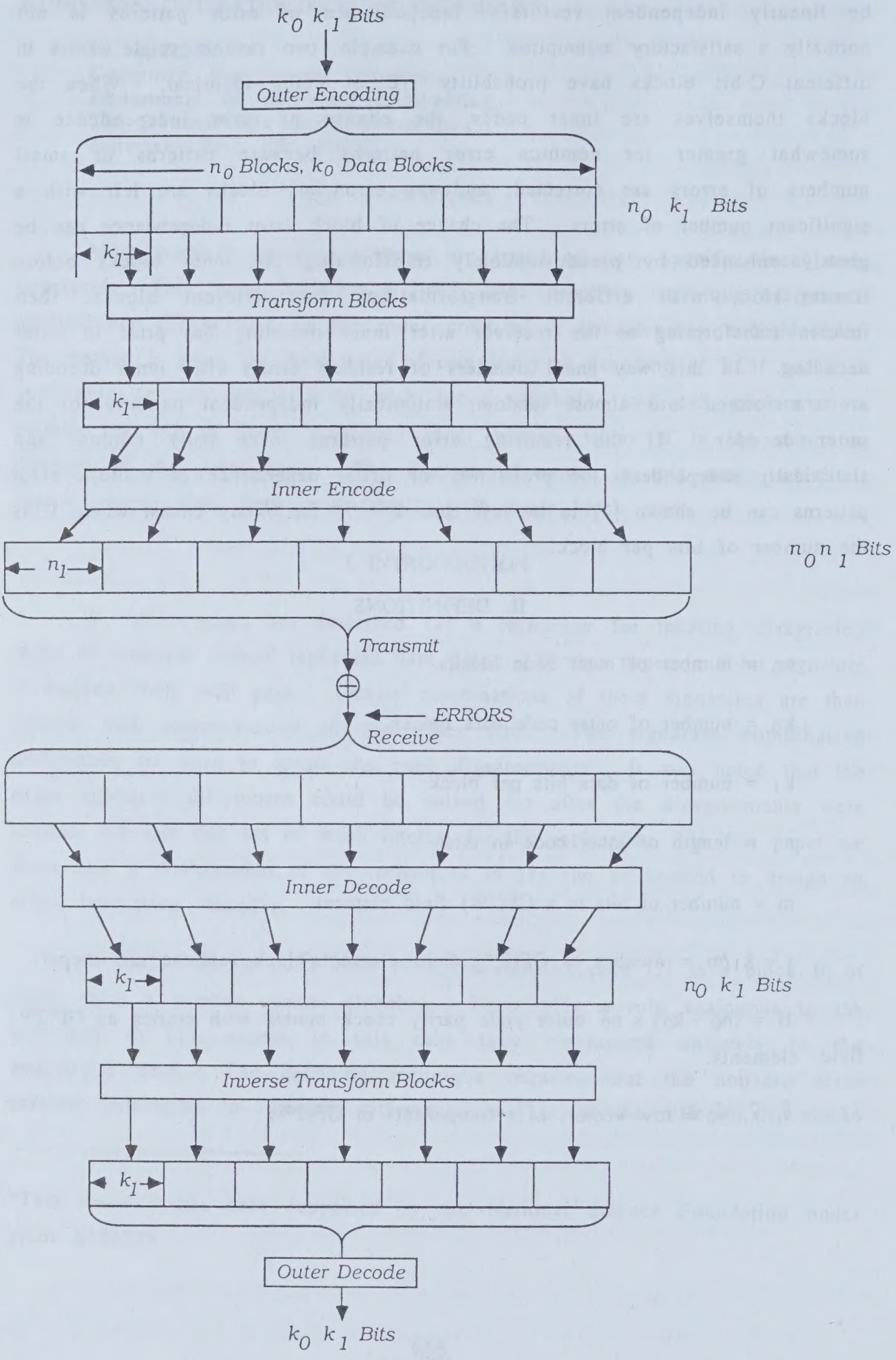
n_1 = length of inner code in bits.

m = number of bits in a $GF(2^m)$ field element.

$r = k_1/m$ = number of $GF(2^m)$ field elements/block (assume an integer).

$H = (n_0 - k_0) \times n_0$ outer code parity check matrix with entries as $GF(2^m)$ field elements.

$\bar{s}_i, \bar{x}_i, \bar{y}_i$ = row vectors of r components in $GF(2^m)$.



$k_0 \; k_1$ Bits
Outer Encoding
n_0 Blocks, k_0 Data Blocks
$n_0 \; k_1$ Bits
k_1
Transform Blocks
k_1
Inner Encode
n_1
$n_0 n_1$ Bits
Transmit
ERRORS
Receive
Inner Decode
k_1
$n_0 \; k_1$ Bits
Inverse Transform Blocks
k_1
Outer Decode
$k_0 \; k_1$ Bits

Figure 1. The Assumed Concatenated Encoding-Decoding Steps.

III. STRUCTURE OF THE ENCODER/DECODER

Figure one illustrates the encoding and decoding steps. First, $k_0 k_1$ bits are encoded into n_0 blocks of k_1 bits each, of which k_0 are data blocks. Each block is then subjected to an invertible transform, with different transformation rules for different blocks. The transformed blocks are then each encoded into an (n_1, k_1) inner code prior to transmission. At the receiver, the (n_1, k_1) blocks are decoded, and n_0 blocks of k_1 bits each are outputted, with some blocks corrected, some not. Each block is then inverse transformed. This has the effect of randomizing the patterns of any remaining errors, so they are more likely to be linearly independent. Outer decoding is then performed.

It is interesting to note that the idea of having different transformations for different inner blocks is reminiscent of Justesen's method [3] of having different codes for different inner blocks of a concatenated code.

IV. OUTER CODE DECODING

Let $\bar{y}_1, \bar{y}_2 \dots \bar{y}_{n_0}$ be the blocks, possibly containing errors, supplied to the outer decoder after inner decoding and inverse transformation.

$$\bar{y}_j = \bar{x}_j + \bar{e}_j,$$

where $\bar{e}_j$ is the error vector.

Let $\bar{s}_1, \bar{s}_2, \dots \bar{s}_{(n_0 - k_0)}$ be syndrome vectors computed by

$$\begin{bmatrix} \bar{s}_1 \\ \bar{s}_2 \\ \cdot \\ \cdot \\ \cdot \\ \bar{s}_{(n_0 - k_0)} \end{bmatrix} = \begin{bmatrix} H \end{bmatrix} \begin{bmatrix} \bar{y}_1 \\ \bar{y}_2 \\ \cdot \\ \cdot \\ \cdot \\ \bar{y}_{(n_0)} \end{bmatrix} = \begin{bmatrix} H \end{bmatrix} \begin{bmatrix} \bar{e}_1 \\ \bar{e}_2 \\ \cdot \\ \cdot \\ \cdot \\ \bar{e}_{(n_0)} \end{bmatrix} \qquad (1)$$

$(n_0 - k_0) \times r$ $\qquad$ $(n_0 - k_0) \times N_0$ $\qquad$ $n_0 \times r$

in abbreviated form,

$$[S] = [H]\ [Y]$$

Suppose there are t nonzero error patterns: $\bar{e}_{j_1}, \bar{e}_{j_2} \ldots . \bar{e}_{j_t}$. Then

$$[S] = \begin{bmatrix} h_{1j_1} & \cdots & h_{1j_t} \\ \vdots & & \\ h_{(n_0-k_0)j_1} & \cdots & h_{(n_0-k_0)j_t} \end{bmatrix} \begin{bmatrix} \bar{e}_{j_1} \\ \bar{e}_{j_2} \\ \vdots \\ \bar{e}_{j_t} \end{bmatrix} \qquad (2)$$

Suppose $\bar{e}j1$, $\bar{e}j2$, ... $\bar{e}jt$ are linearly independent. Of necessity, $r \geq t$. If we take linear combinations of the $n_0 - k_0$ row equations in matrix equation (1), some of the equations will not include $e_{j_1}, e_{j_2}, \ldots e_{j_t}$, and will have the right hand side equal to zero. Call these null combinations. If an equation does include any of the $\bar{e}_{j_1}, \bar{e}_{j_2}, \ldots \bar{e}_{j_t}$, it will not be a null combination because of the linear independence of $\bar{e}_{j_1}, \bar{e}_{j_2}, \ldots \bar{e}_{j_t}$. Some results were proved in [1] that, although interpreted somewhat differently there, translate to the following in terms of the codes considered in this paper. What was considered the check digits in [1] is now the syndrome in this paper. What was called the data set in [1] is here the whole frame: n_o blocks of k_1 bits each.

1. If every set of t+1 columns of H is linearly independent, and if there are t errors, then every correct block will be present in at least one null combination. (Actually, it was shown for the binary case that each correct block appears in exactly half the null combinations, and it is easily shown that, in the general case $q = 2^m$, it appears in $(q-1)/q$ of them).

2. If the outer code has minimum distance d, any pattern of d-2 or fewer errors can be located if the error patterns are linearly independent.

3. If there are t errors, and $t \leq d-2$, all the correct positions can be found by inspection of at most $n_0 - k_0 - t$ null combinations, which are linearly independent equations.

4. Once located, equation (2) can be solved for the exact error patterns.

The decoding steps are basically the same as the page disagreement location in [1]. They are as follows:

1. Suppose there are $t \leq d-2$ linearly independent error patterns. The rank of [S] will be t. Perform a Gauss-Jordan reduction on [S] using only column operations, obtaining t independent rows with a single one. The decoder now knows there are t block errors. The other $n_0 - k_0 - t$ rows will reveal directly the $n_0 - k_0 - t$ linearly independent null combinations.

2. Select the first null combination, and take the same combination of rows of H. Any nonzero position in the combination of rows of H is now known to be a correct block position. On average, (q-1)/q of the correct blocks will be found at this first step. Continue taking null combinations until n_0-t blocks have been found correct as a result of appearing in at least one null combination. This will take at most n_0-k_0-t trials, but usually will take much fewer.

3. After the positions of the erroneous blocks have been found, invert a t x t submatrix of H consisting of the t columns of H corresponding to the error positions, and the t rows of H corresponding to the t rows containing a single one found in the Gauss-Jordan reduction. This inverted matrix can multiply the appropriate t linearly independent syndrome vectors to obtain the error patterns.

As to the complexity of decoding, there are interesting parallels with BCH decoding [4]. In BCH decoding, an error locating polynomial must be found, which involves inversion of a t x t matrix; the Gauss Jordan reduction to find the rank and null combinations also is approximately equivalent to a t x t matrix inversion, and the null combinations are analogous to the error locating polynomial. To find the error values, both procedures must invert another t x t matrix.

V. AN ILLUSTRATION USING A REED-SOLOMON CODE

Consider a case of $n_0 = 7$ blocks, $t = 3$ erroneous blocks and $k_0 = 3$ data blocks. The (7,3) Reed-Solomon code has $d = 5$. Its parity check matrix can be constructed from h(x) over GF(8). α is a primitive element of the field, represented by polynomials over GF(2) modulo $x^4 + x + 1$.

$$h(x) = (x - \alpha^5)(x - \alpha^6)(x-1) = x^3 + \alpha^3 x^2 + \alpha^2 x + \alpha^4$$

$$[H] = \begin{bmatrix} 0 & 0 & 0 & 1 & \alpha^3 & \alpha^2 & \alpha^4 \\ 0 & 0 & 1 & \alpha^3 & \alpha^2 & \alpha^4 & 0 \\ 0 & 1 & \alpha^3 & \alpha^2 & \alpha^4 & 0 & 0 \\ 1 & \alpha^3 & \alpha^2 & \alpha^4 & 0 & 0 & 0 \end{bmatrix}$$

Let positions 1, 4, and 5 be in disagreement, where the blocks are numbered from 0 through 6 from left to right. The block size as a minimum needs to be 3·3=9 bits, and should be substantially larger to reduce the chance of error pattern linear dependence. However, we will assume for simplicity a 9-bit block, and select the three errors to be linearly independent as follows.

Block Position	Error Pattern
1	$[1\ 1\ \alpha]$
4	$[\alpha^2 \alpha 1]$
5	$[\alpha^3 0 \alpha]$

Computation according to (1) yields

$$[S] = \begin{bmatrix} 0 & \alpha^4 & 0 \\ \alpha^5 & \alpha^3 & \alpha^3 \\ \alpha^2 & \alpha^4 & \alpha^2 \\ \alpha^3 & \alpha^3 & \alpha^4 \end{bmatrix}$$

After Gauss-Jordan column operations we obtain:

$$[S]' = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \alpha^6 & \alpha^4 & \alpha^3 \end{bmatrix}$$

The rank is three, which means there are at least three errors. If the blocks were larger, as they should be, the existence of a greater number of disagreements than three would usually yield a rank 4 result, indicating 4 or more disagreements.

Given that there are actually three errors, [S]' tells us exactly which combination of the rows of H is a null combination. Note that in this case $n_0 - k_0 - t = 1$, so there is only one null combination. It is α^6 times the first row plus α^4 times the second row plus α^3 times the third row plus the fourth row. This can be expressed alternatively in terms of a polynomial times h(x):

$$(\alpha^6 + x\alpha^4 + x^2\alpha^3 + x^3)\ h(x)$$

Forming this combination:

$$\begin{array}{ccccccccc}
 & 0 & 0 & 0 & \alpha^6 & \alpha^2 & \alpha & \alpha^3 \\
+ & 0 & 0 & \alpha^4 & 1 & \alpha^6 & \alpha & 0 \\
+ & 0 & \alpha^3 & \alpha^6 & \alpha^5 & 1 & 0 & 0 \\
+ & 1 & \alpha^3 & \alpha^2 & \alpha^4 & 0 & 0 & 0 \\
\hline
 & 1 & 0 & \alpha^5 & \alpha^6 & 0 & 0 & \alpha^3
\end{array}$$

The three resulting zero values reveal the disagreements in blocks 1, 4, and 5. Note that the combination and error locations can be found in a very simple manner: just take the elements found from the Gauss-Jordan reduction - α^6, α^4, α^3 and 1 - and shift them into a finite impulse response shift register, which multiplies by h(x) and record the zero output positions: these are the error locations.

To find the error values, we invert columns 1, 4 and 5 and rows 1, 2 and 3 of H:

$$\begin{bmatrix} 0 & \alpha^3 & \alpha^2 \\ 0 & \alpha^2 & \alpha^4 \\ 1 & \alpha^4 & 0 \end{bmatrix} \xrightarrow{\text{Inversion}} \begin{bmatrix} \alpha^3 & \alpha & 1 \\ \alpha^6 & \alpha^4 & 0 \\ \alpha^4 & \alpha^5 & 0 \end{bmatrix}$$

Then multiply the inverted part of H by the S matrix to obtain the three error patterns as the three rows of the result.

$$\begin{bmatrix} \alpha^3 & \alpha & 1 \\ \alpha^6 & \alpha^4 & 0 \\ \alpha^4 & \alpha^5 & 0 \end{bmatrix} \begin{bmatrix} 0 & \alpha^4 & 0 \\ \alpha^5 & \alpha^3 & \alpha^3 \\ \alpha^2 & \alpha^4 & \alpha^2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & \alpha \\ \alpha^2 & \alpha & 1 \\ \alpha^3 & 0 & \alpha \end{bmatrix}$$

VI. LINEAR INDEPENDENCE AND BLOCK TRANSFORMATIONS

The fraction of all sets of t error patterns that are linearly dependent is, from reference [1], approximately

$$2^{-m(r-t+1)}$$

This falls exponentially with the number, r, of GF(2^m) symbols in an inner block. This fraction is not affected by the block transformations. The reason for the block transformations is that within certain common classes of error patterns the fraction of dependent cases is much larger. The case of single errors having probability $1/k_1 = 1/mr$ of being identical was mentioned previously. However, single errors will not exist after the inner decoding. Another common case is two identical multiple error patterns in two different blocks. By having different transformations for different blocks, these are transformed into different error patterns. However, it still might be necessary to worry about some other common patterns such as an error pattern in one block being a cyclic shift of a pattern in another block.

Selection of the transformation rules involves a tradeoff between ideal completely randomized transformations and the desire to keep the transformation rule simple. In this paper we will simply make some observations about transformation rules, but will not attempt to further quantize or optimize the choices.

One transformation which comes to mind is to multiply the ith block by α^i, where α is a primitive element of GF(2^{mr}). The block data is treated as a field element in the multiplication. A potential problem with this could be that, if $\alpha = x$ is the primitive element in a field module p(X), an error pattern in block i that was a j-step cyclic shift of an error pattern in block i+j could come out identical to the other after the transformation. This might be avoided by multiplying by $\beta + \alpha^i$, where β is the same for all blocks.

CONCLUSIONS

The outer code decoding technique which has been described has several interesting features. First, the outer code error correcting capability is limited to d-2 instead of the usual d/2. Although not all patterns of d-2 or fewer inner block errors are correctable, the fraction which is not goes down exponentially with inner block length. Second, the decoding technique applies to any linear parity check code. Codes with best minimum distance can be selected without much other concern about their structure. Third, the decoding technique is of about the same order of complexity as BCH codes, possibly simpler in some circumstances. Certainly the concept is easier to explain (Compare the 17 pages in reference [4] needed to explain a BCH decoder with the 3 steps listed in the Outer Code Decoding section of this paper). One element of complexity which is not present in ordinary BCH decoding is the need to transform and inverse transform the inner data blocks. However, these are fixed transform rules, which can be hard wired or in ROM, and the blocks can be processed in parallel to a large degree.

REFERENCES

[1] E. J. Kapturowski and J.J. Metzner, "Error-Location Codes - Error Control Through Comparison of Replicated Files", submitted for publication. Based in part on Thesis by E.J. Kapturowski for M.S. in Computer Engineering at Pennsylvania State University, June 1988.

[2] G.D. Forney, Jr., Concatenated Codes. M.I.T. Press, Cambridge, Mass, 1966.

[3] J. Justesen, "A Class of Constructive Asymptotically Good Algebraic Codes", IEEE Transactions on Information Theory, Vol. IT-18, pp. 652-656, September 1972.

[4] R.E. Blahut, Theory and Practice of Error Control Codes, Addison Wesley, 1983, pp. 166-183.

AN IMPROVED DECODING ALGORITHM FOR CERTAIN ELLIPTIC CODES

CHAO-MING LIU and P. VIJAY KUMAR
Department of Electrical Engineering-Systems
University of Southern California
Los Angeles, CA 90089-0272

ABSTRACT

Recently, Driencourt constructed a family of (algebraico-geometric) Goppa codes on elliptic curves and outlined a decoding algorithm for these codes. Here, we show how the multisequence shift-register synthesis algorithm of Feng and Tzeng can be extended to the case when the individual sequences have different lengths and then apply the extended algorithm, to provide an improved decoding algorithm for these elliptic codes. Using shift-register synthesis, all error patterns in which the number of errors does not equal or exceed $\lfloor \frac{d_{\min}}{3} \rfloor$ are guaranteed to be corrected.

1. INTRODUCTION

In [4], Driencourt constructs a family of elliptic codes ((D, G) Goppa codes[1] on elliptic curves[6]) whose parity-check matrix has a simple structure and outlined a decoding algorithm for these codes which is guaranteed to correct all error patterns in which the number of errors does not equal or exceed $\frac{d}{4}$, d being the minimum distance of the elliptic code.

Feng and Tzeng[5] recently presented a generalized Euclidean division algorithm solving the difficult problem of finding the shortest-length linear shift-register for m sequences of the same length N.

In this paper, we first show that when the error pattern in an elliptic code is of a certain type, the decoding of Driencourt's elliptic code reduces to the task of finding the least-degree common linear recursion satisfied by two sequences (of different lengths) derived from the syndrome vector. We then show how the shift-register synthesis algorithm of Feng and Tzeng can be adapted to handle the case when the individual sequences have different lengths. The resulting decoding algorithm improves over the algorithm of Driencourt and is guaranteed for instance, to correct all error patterns in which the number of errors does not equal or exceed $\lfloor \frac{d}{3} \rfloor$. It is also easy to implement as the shift-register synthesis algorithm is based upon repeated use of the Euclidean division algorithm.

2. DRIENCOURT'S ELLIPTIC CODES

Rational points on an elliptic curve C defined by a Weierstrass polynomial[4] other than the point at infinity can be grouped into pairs of the form

$$P_i = [\alpha_i : \beta_i : 1], \quad \overline{P}_i = [\alpha_i : \gamma_i : 1],$$

with one exception when the characteristic of the field is 2. The points P_i and $\overline{P}_i$ may be called *opposite* points.

Set $G = dP_\infty$ ($P_\infty = [0 : 1 : 0]$) for some positive integer d and let $\delta = \lfloor \frac{d}{2} \rfloor$. Using the fact that

$$(z^\delta) = 3\delta P_\infty = dP_\infty + (3\delta - d)P_\infty$$

Driencourt showed that to obtain a basis for $L(G)$ it suffices to consider the rational functions

$$1, \frac{x}{z}, \ldots, \frac{x^\delta}{z^\delta}, \frac{y}{z}, \frac{yx}{z^2}, \ldots, \frac{yx^{\delta'}}{z^{\delta'+1}},$$

where $\delta' = \lfloor \frac{d-3}{2} \rfloor$. Then the parity-check matrix of the (D, G) code is of the form

$$H = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 & 1 & \cdots & 1 & 1 \\ \alpha_0 & \alpha_1 & \alpha_1 & \cdots & \alpha_i & \alpha_i & \cdots & \alpha_s & \alpha_s \\ \vdots & \vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots \\ \alpha_0^\delta & \alpha_1^\delta & \alpha_1^\delta & \cdots & \alpha_i^\delta & \alpha_i^\delta & \cdots & \alpha_s^\delta & \alpha_s^\delta \\ \beta_0 & \beta_1 & \gamma_1 & \cdots & \beta_i & \gamma_i & \cdots & \beta_s & \gamma_s \\ \beta_0\alpha_0 & \beta_1\alpha_1 & \gamma_1\alpha_1 & \cdots & \beta_i\alpha_i & \gamma_i\alpha_i & \cdots & \beta_s\alpha_s & \gamma_s\alpha_s \\ \vdots & \vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots \\ \beta_0\alpha_0^{\delta'} & \beta_1\alpha_1^{\delta'} & \gamma_1\alpha_1^{\delta'} & \cdots & \beta_i\alpha_i^{\delta'} & \gamma_i\alpha_i^{\delta'} & \cdots & \beta_s\alpha_s^{\delta'} & \gamma_s\alpha_s^{\delta'} \end{bmatrix}, \qquad (1)$$

where $s = \lfloor \frac{n}{2} \rfloor$. Let the error vector corresponding to the above parity-check matrix be

$$\underline{e} = [e_0, e_1, \overline{e}_1, \cdots, e_i, \overline{e}_i, \cdots, e_s, \overline{e}_s]^t.$$

With a little manipulation, the syndrome can be broken into two vectors $\underline{S}$ and $\underline{T}$ satisfying

$$\underline{S} = \begin{bmatrix} S_0 \\ S_1 \\ \vdots \\ S_\delta \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \alpha_0 & \alpha_1 & \cdots & \alpha_s \\ \vdots & \vdots & & \vdots \\ \alpha_0^\delta & \alpha_1^\delta & \cdots & \alpha_s^\delta \end{bmatrix} \begin{bmatrix} \epsilon_0 \\ \epsilon_1 \\ \vdots \\ \epsilon_s \end{bmatrix} \tag{2}$$

and

$$\underline{T} = \begin{bmatrix} T_0 \\ T_1 \\ \vdots \\ T_{\delta'} \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \alpha_0 & \alpha_1 & \cdots & \alpha_s \\ \vdots & \vdots & & \vdots \\ \alpha_0^{\delta'} & \alpha_1^{\delta'} & \cdots & \alpha_s^{\delta'} \end{bmatrix} \begin{bmatrix} \eta_0 \\ \eta_1 \\ \vdots \\ \eta_s \end{bmatrix}, \tag{3}$$

where we introduce a fictitious $\overline{e}_0$ (always equal to zero) and define

$$\epsilon_i = e_i + \overline{e}_i, \quad \eta_i = \beta_i e_i + \gamma_i \overline{e}_i, \quad \forall\, i, \quad 0 \le i \le s.$$

It is clear that the determination of ϵ_i and η_i leads to that of e_i and $\overline{e}_i$. Driencourt's approach was to solve (2) and (3) separately for the common error locator polynomial.

3. A NEW DECODING ALGORITHM

What is not clear however, is that this common recursion is the *unique least-degree* commmon linear recursion of the sequences $\underline{S}$ and $\underline{T}$. After some work, we are able to identify a certain condition under which this is true and then show how this fact can be used to determine the error locator polynomial.

We restrict our attention to the case $d_{\min} = d = \delta + \delta' + 2$ (the lesser of the two possible values), in which case the code is only guaranteed to correct at most $\delta' + 1$ errors. Thus we may assume $\deg \sigma(z) \le \delta' + 1$ and $\deg w_i(z) \le \delta'$, $i = 1, 2$.

Let s (for single error) denote the number of pairs in which either $e_i \ne 0$ or $\overline{e}_i \ne 0$ but not both. Let t (for two errors) denote the number of pairs such that both $e_i \ne 0$ and $\overline{e}_i \ne 0$. Then

Theorem 1 *If* $3s + 4t \le \delta + \delta' + 1$, *then the characteristic polynomial of the least degree linear recursion generating* $\underline{S}$ *and* $\underline{T}$ *is unique and is in fact the error locator polynomial* $\sigma(z)$.

Thus, when $3s + 4t \le \delta + \delta' + 1$, the error locator polynomial can always be found by finding the shortest-length linear shift-register generating the two sequences $\underline{S}$ and $\underline{T}$.

In [5], Feng and Tzeng present a generalized Euclidean division algorithm solving the difficult problem of finding the shortest-length linear-feedback shift-register that can generate a given set of m sequences *having the same length N*. The algorithm can be extended to the case when the sequences have different lengths. This is done by adjoining zeros to the end of the sequences of shorter length to make all sequences have equal length and defining $b_0^{(h)}(z) = z^{mN_h + h}$. This is a necessary step as the two "halves" $\underline{S}$ and $\underline{T}$ of the syndrome have different lengths. Due to lack of space, we do not expand on this here and the generalization of this decoding algorithm to other codes is also omitted.

References

[1] V. D. Goppa, "Algebraico-geomrtric codes," *Math. USSR Izvestiya*, vol.21, pp. 75–91, 1983.

[2] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-correcting Codes*, New York: North Holland, 1983.

[3] J. W. P. Hirschfeld, "Linear codes and algebraic curves," in *Geometrical Combinatorics*, F. C. Holroyd and R. J. Wilson, Eds., Boston, MA:Pitman, 1984.

[4] Y. Driencourt, "Some properties of elliptic codes over a field of characteristic 2," *Proceedings of the 3rd International Conference, AAECC-3*, Grenoble, France, pp. 185–193, July 1985.

[5] G. L. Feng and K. K. Tzeng, "A generalized Euclidean algorithm for multisequence shift-register synthesis," presented in part at the *Fourth Carribean Conference on Combinatorics and Computing*, University of Puerto Rico, Rio Piedras, Puerto Rico, April 1985, and at the *IEEE International Symposium on Information Theory*, Brighton, England, June 1985.

[6] W.Fulton, *Algebraic Curves*, Math. Lecture Notes Ser., Benjamin, 1969.

OPTICAL ORTHOGONAL CODES – NEW BOUNDS AND AN OPTIMAL CONSTRUCTION

HABONG CHUNG & P. VIJAY KUMAR
Department of Electrical Engineering-Systems,
University of Southern California,
Los Angeles, CA 90089-0272.

ABSTRACT

Recently Chung, Salehi, and Wei defined an (n, ω, λ)-optical orthogonal code (OOC) to be a family of (0,1)-sequences of length n and constant Hamming-weight ω which have both out-of-phase autocorrelation and crosscorrelation values not exceeding λ. Such codes find application in optical code-division multiple-access communication systems.

Such an $(n.\omega, \lambda)$-OOC is closely related to a "cyclic" Steiner system $s(t, k, v)$, in which $t = \lambda + 1$, $k = \omega$, and $v = n$. Most known optimal constructions have thus far been for the case $\lambda = 1$.

In this paper, a technique for constructing optimal OOC's is presented. This technique uses finite field theory and provides the only known family of optimal (with respect to family size) OOC's having $\lambda = 2$. The parameters (n, ω, λ) are respectively $(p^{2m} - 1, p^m + 1, 2)$, where p is any prime.

It is clearly of theoretical and practical interest to estimate the maximum size of an OOC having specified parameters (n, ω, λ). Three distinct upper bounds on the size of an OOC are presented. The first bound identifies values of the parameter (n, ω, λ) for which only a trivial OOC (family size ≤ 1) can exist. The second and third bounds both improved upon the tightest previously known bound (based on the Johnson bound) for many values of the parameter set (n, ω, λ). Both are based on an estimation of the higher-order correlation moments of the code.

1. INTRODUCTION

An (n, ω, λ)-optical orthogonal code (OOC) [1-2] C is a family of (0,1)-sequences of length n and Hamming-weight ω satisfying the following two properties:

(a) the autocorrelation property

$$\sum_{k=0}^{n-1} x(k)x(k \oplus \tau) \leq \lambda \tag{1}$$

$\forall$ sequences $x(\cdot) \in C$ and all integers $\tau \not\equiv 0 \pmod{n}$.

(b) the crosscorrelation property

$$\sum_{k=0}^{n-1} x(k)y(k \oplus \tau) \leq \lambda \tag{2}$$

$\forall$ pairs of distinct sequences $x(\cdot)$, $y(\cdot)$ in C, and all integers τ. Note that in (1) and (2), $\oplus$ denotes addition modulo n.

Such codes find application in optical code-division multiple-access communication systems.

For a given set of values of n, ω, and λ, the largest possible size of an (n, ω, λ)-optical orthogonal code is denoted by $\Phi(n, \omega, \lambda)$. A code achieving this maximum size is termed *optimal.* Since it is difficult to determine the exact value of $\Phi(n, \omega, \lambda)$ in general, upper bounds on $\Phi(n, \omega, \lambda)$ are of interest.

In this paper, we present three distinct upper bounds on the size of an optical orthogonal code in sections 2,3, and 4, respectively. The first bound identifies values of the parameters (n, ω, λ) for which only a trivial OOC (family size ≤ 1) can exist. The second and third bounds both improve upon the bound in (3) (based on the Johnson bound) on family size for many values of the parameter set (n, ω, λ). Both are based on an estimation of the higher-order correlation moments of the code. In the last section, Section 5, a new (in fact the only known) construction for an optimal OOC having $\lambda = 2$ is presented. (In all the systematic construction methods presented in [1] and [2], the optimal OOC's constructed turn out to have the parameter $\lambda = 1$.) For the sake of brevity, we omit the proofs of all theorems and corollaries.

2. A SIMPLE UPPER BOUND ON FAMILY SIZE

It is easy to show that the codewords in an (n, ω, λ)-optical orthogonal code, together with all their cyclic shifts form a constant-weight error-correcting code of length n, weight ω, and minimum distance $2(\omega - \lambda)$. Therefore, an upper bound on the maximum possible size $A(n, \omega, \lambda)$ of a constant-weight error-correcting code of length n, weight ω, and minimum distance $2(\omega - \lambda)$ can be transformed (by dividing by the code length) into an upper bound on the size $\Phi(n, \omega, \lambda)$ of an (n, ω, λ)-optical orthogonal code. The most general upper bound for $\Phi(n, \omega, \lambda)$ contained in [1] is derived from the Johnson bound for constant-weight codes and states that

$$\Phi(n, \omega, \lambda) \leq \left\lfloor \frac{1}{\omega} \left\lfloor \frac{(n-1)}{(\omega-1)} \left\lfloor \frac{(n-2)}{(\omega-2)} \left\lfloor \cdots \left\lfloor \frac{(n-\lambda)}{(\omega-\lambda)} \right\rfloor \right\rfloor \cdots \right\rfloor \right\rfloor \right\rfloor . \tag{3}$$

This bound may be derived using the recursion

$$A(n, \omega, \lambda) \leq \left\lfloor \frac{n}{\omega} \cdot A(n-1, \omega-1, \lambda-1) \right\rfloor \tag{4}$$

and the obvious bound

$$A(n-\lambda, \omega-\lambda, 0) \leq \left\lfloor \frac{n-\lambda}{\omega-\lambda} \right\rfloor . \tag{5}$$

Clearly, the bound in (5) is as tight as possible and one therefore expects the bound in (3) to be tight for small values of λ.

The cyclic-correlation properties required of an OOC lead to the two simple bounds given below:

Theorem 1

$$\begin{aligned} &1.\ \Phi(n, \omega, \lambda) &\leq\ & 1 \quad \text{if} \quad \omega^2 > n\lambda && (6) \\ &2.\ \Phi(n, \omega, \lambda) &=\ & 0 \quad \text{if} \quad \omega(\omega-1) > (n-1)\lambda. && (7) \end{aligned}$$

Proof: (omitted).

Thus our interest lies chiefly in those sets (n, ω, λ) in which $\omega^2 \leq n\lambda$.

3. AN UPPER BOUND BASED UPON HIGHER-ORDER CORRELATION MOMENTS.

The higher-order correlation moments have been employed in the past to derive a lower bound on the maximum correlation value of a family of fixed-energy *complex* sequences (see Welch [3]). However, it is possible to do better than the Welch bound by using the cyclic-correlation constraints imposed on an OOC as well as the fact that the symbols can only take on the values 1 or 0. The bound on maximum family size resulting from an examination of the l^{th}-order moments, $l \geq 1$, is given below in Theorem 2. The bound in Theorem 2 can be derived in the following two steps. We first obtain separately, a lower and an upper bound on the l^{th}-order moment m_l in terms of the maximum correlation parameter λ. We then relate the two bounds to obtain a bound on the maximum possible size M of a family.

Theorem 2

$$\Phi(n, \omega, \lambda) \leq \frac{\omega(\omega^{l-1} - \lambda^{l-1})}{n\left\{\sum_{u=1}^{l} C_{l,u} \binom{\omega}{u}^2 / \binom{n}{u}\right\} - \omega^2 \lambda^{l-1}} \qquad \text{if } \lambda \geq l, \tag{8}$$

and

$$\Phi(n, \omega, \lambda) \leq \frac{\omega(\omega^{l-1} - \lambda^{l-1}) - \sum_{u=\lambda+1}^{l} C_{l,u} \binom{\omega}{u}}{n\left\{\sum_{u=1}^{\lambda} C_{l,u} \binom{\omega}{u}^2 / \binom{n}{u}\right\} - \omega^2 \lambda^{l-1}} \qquad \text{if } \lambda < l. \tag{9}$$

where

$$C_{l,u} = \sum_{i=0}^{u} (-1)^i \binom{u}{i} (u-i)^l .$$

Proof: (omitted).

It often happens that for many choices of parameters n, ω, λ (especially for large λ), the denominator of the RHS of (8) or (9) becomes negative and the bound can not be applied in this case. However, for other values of parameters n, ω, and λ, and a proper choice of l, the bound in Theorem 2 is significantly tighter than the Johnson bound.

4. A MOMENT-BASED BOUND THAT CAN BE EXTENDED RECURSIVELY

It is not difficult to see that the bound in Theorem 2 can be modified in a straightforward fashion to yield a bound on the size of a constant-weight error-correcting code having given minimum distance $2(\omega - \lambda)$. This bound is stated in Corollary 1.

Corollary 1

Let $A(n, \omega, \lambda)$ be the maximum number of codewords in a constant-weight error-correcting code in which the codewords have length n, weight ω, and minimum distance $2(\omega - \lambda)$. Then,

$$A(n,\omega,\lambda) \leq \frac{\omega^l - \lambda^l}{\left\{\sum_{u=1}^{l} C_{l,u}\binom{\omega}{u}^2/\binom{n}{u}\right\} - \lambda^l} \qquad \text{if } \lambda \geq l, \tag{10}$$

and

$$A(n,\omega,\lambda) \leq \frac{\left\{\sum_{u=1}^{\lambda} C_{l,u}\binom{\omega}{u}\right\} - \lambda^l}{\left\{\sum_{u=1}^{\lambda} C_{l,u}\binom{\omega}{u}^2/\binom{n}{u}\right\} - \lambda^l} \qquad \text{if } \lambda < l. \tag{11}$$

Proof: (omitted).

By using the inequality

$$\Phi(n,\omega,\lambda) \leq \frac{A(n,\omega,\lambda)}{n},$$

one obtains a bound on the size of an OOC, namely

$$\Phi(n,\omega,\lambda) \leq \frac{\omega^l - \lambda^l}{n\left[\left\{\sum_{u=1}^{l} C_{l,u}\binom{\omega}{u}^2/\binom{n}{u}\right\} - \lambda^l\right]} \qquad \text{if } \lambda \geq l, \tag{12}$$

$$\Phi(n,\omega,\lambda) \leq \frac{\left\{\sum_{u=1}^{\lambda} C_{l,u}\binom{\omega}{u}\right\} - \lambda^l}{n\left[\left\{\sum_{u=1}^{\lambda} C_{l,u}\binom{\omega}{u}^2/\binom{n}{u}\right\} - \lambda^l\right]} \qquad \text{if } \lambda < l. \tag{13}$$

When the bound in Theorem 2 applies i.e., when for a fixed value of n, ω, λ and l, the denominator of either (8) (when $\lambda \geq l$) or (9) (when $\lambda < l$) is positive, the resultant bound is tighter than that obtained from equations (12) and (13). However, the latter has the advantage that its range of applicability can be increased by recursion as in the case of the Johnson bound. Such an extension of the above bound is presented below.

Theorem 3

If $\lambda - d \geq l$,

$$\Phi(n,\omega,\lambda) \leq \left\lfloor \frac{1}{\omega}\left\lfloor \frac{(n-1)}{(\omega-1)}\left\lfloor \cdots \left\lfloor \frac{(n-d+1)}{(\omega-d+1)}\left\lfloor \frac{(\omega')^l - (\lambda')^l}{\left\{\sum_{u=1}^{l} C_{l,u}\binom{\omega'}{u}^2/\binom{n'}{u}\right\} - (\lambda')^l}\right\rfloor\right\rfloor \cdots \right\rfloor\right\rfloor\right\rfloor,$$

and if $\lambda - d < l$,

$$\Phi(n,\omega,\lambda) \leq \left\lfloor \frac{1}{\omega}\left\lfloor \frac{(n-1)}{(\omega-1)}\left\lfloor \cdots \left\lfloor \frac{(n-d+1)}{(\omega-d+1)}\left\lfloor \frac{\left\{\sum_{u=1}^{\lambda'} C_{l,u}\binom{\omega'}{u}\right\} - (\lambda')^l}{\left\{\sum_{u=1}^{\lambda'} C_{l,u}\binom{\omega'}{u}^2/\binom{n'}{u}\right\} - (\lambda')^l}\right\rfloor\right\rfloor \cdots \right\rfloor\right\rfloor\right\rfloor.$$

where $n' = n - d$, $\omega' = \omega - d$, and $\lambda' = \lambda - d$.

Proof: (omitted).

By choosing a suitable value of d and l, we can often obtain an upper bound for some values of n for which the bound in Theorem 2 fails to apply. Also, even if Theorem 2 does apply, this method (Theorem 3) many times provides an improved bound. For example, if $\omega = 20$ and $\lambda = 3$, Theorem 2 applies (i.e., there exists some value of l for which the bound can be applied) only as long as $n \leq 243$, whereas Theorem 3 can be applied provided $n \leq 325$. Table 1 compares the Johnson bound, the bound in Theorem 2, and the above recursive bound for several values of the parameters n, ω, and λ. In Table 1, B is a reference to the Johnson bound, and B_l refers to the bound in Theorem 2 obtained using the l^{th}-order correlation moment, whereas $R_{d,l}$ refers to the bound in Theorem 3 using the specified d and l.

ω	λ	n	B	Bound in Th. 2	Bound in Th. 3
10	2	55	3	$B_{10} = 1$	$R_{1,9} = 1$
		60	4	$B_{10} = 2$	$R_{1,9} = 2$
		70	6	$B_{10} = 4$	$R_{1,9} = 4$
		80	7	$B_{10} = 31$	$R_{1,9} = 31$
		90	10	$*$	$*$
	3	35	6	$B_2 = 1$	$R_{1,9} = 1$
		40	9	$B_{10} = 4$	$R_{2,8} = 4$
		50	19	$B_{10} = 38$	$R_{2,8} = 10$
		60	38	$*$	$R_{2,8} = 43$
		70	58	$*$	$*$
20	2	250	8	$B_{20} = 2$	$R_{1,19} = 2$
		300	12	$B_{20} = 4$	$R_{1,19} = 4$
		350	17	$B_{20} = 26$	$R_{1,19} = 26$
	3	150	25	$B_2 = 2$	$R_{1,18} = 2$
		200	63	$B_{20} = 17$	$R_{2,18} = 13$
		250	125	$*$	$R_{2,18} = 36$
		300	221	$*$	$R_{2,18} = 152$
		350	354	$*$	$*$
	4	120	98	$B_3 = 7$	$R_{2,17} = 6$
		150	248	$B_{20} = 65$	$R_{2,17} = 32$
		200	800	$*$	$R_{3,17} = 195$
		250	1958	$*$	$R_{3,17} = 848$
		300	4090	$*$	$*$

$*$ *bound not applicable*

Table 1: Comparison of the Bounds

S_1:	$\{1, 5, 8, 18, 28, 31, 35, 40, 59\}$
S_2:	$\{2, 7, 10, 16, 17, 36, 55, 56, 62\}$
S_3:	$\{3, 11, 24, 25, 27, 29, 30, 43, 51\}$
S_4:	$\{4, 9, 14, 20, 32, 34, 47, 49, 61\}$
S_5:	$\{6, 22, 23, 39, 48, 50, 54, 58, 60\}$
S_6:	$\{12, 15, 33, 37, 44, 45, 46, 53, 57\}$

Table 2: Codeword-sets of (63,9,2)-OOC

5. AN OPTIMAL ($\lambda = 2$) CONSTRUCTION

Using some simple finite field theory, we present below a new (in fact the only known) and optimal family of optical orthogonal codes having $\lambda = 2$. Again, for lack of space, we omit the proof.

Construction 1

Let α be a primitive element of $\mathrm{GF}(p^{2m})$, p any prime. Set $q = p^m + 1$, and let $\beta = \alpha^q$. Let f be the (logarithmic) map from $\mathrm{GF}(p^{2m}) \setminus \{0\}$ to the integers modulo $p^{2m} - 1$ given by:

$$f(\alpha^t) = t.$$

Then the characteristic functions of the sets S_i, $1 \le i \le p^m - 2$, given by

$$S_i = \{\, f(x) \mid (x-1)^{p^m+1} = \beta^i \,\}, \tag{14}$$

may be taken as codewords of an optimal $(p^{2m} - 1, p^m + 1, 2)$-optical orthogonal code.

In Table 2, codeword-sets of an optimal $(2^6 - 1, 2^3 + 1, 2)$-optical orthogonal code constructed by Construction 1 are listed.

References

[1] F. R. K. Chung, J. A. Salehi, V. K. Wei, "Optical Orthogonal Code: Design, Analysis, and Application," presented at 1986 IEEE International Symposium on Information Theory, Ann Arbor, Michgan.

[2] E. F. Brickell, and V. K. Wei, "Optical Orthogonal Codes and Cyclic Block Designs," *Congressus Numerantium*, **58**, pp. 175-192, 1987.

[3] L. R. Welch, "Lower Bounds on the Maximum Cross Correlation of Signals," *IEEE Transactions on Information Theory*, vol. IT-20, pp. 397-399, May 1974.

[4] H. Hanani. "The Existence and Constructions of Balanced Incomplete Block Designs," *Annals Math. Stat.*, **32**, pp. 361-386, 1961.

[5] R. C. Bose. "On the Construction of Balanced Incomplete Block Designs," *Annals Eugenics*, **9**, pp. 353-399, 1939.

[6] R. M. Wilson. "Cyclotomy and Difference Families in Elementary Abelian Groups," *Journal of Number Theory*, **4**, pp. 17-47, 1972.

[7] F. J. MacWilliams, N. J. A. Sloane. *The Theory of Error-Correcting Codes*, Amsterdam; North-Holland Publishing Company, 1977.

[8] G. Polya, R. E. Tarjan, D. R. Woods. *Notes on Introductory Combinatorics*, Birkhauser Boston, Inc., 1983

MAXIMUM MUTUAL INFORMATION DECODING FOR ARBITRARILY VARYING MULTIPLE-ACCESS CHANNELS

JOHN A. GUBNER[1]
Department of Electrical and Computer Engineering
University of Wisconsin–Madison
Madison, WI 53706, U.S.A.

ABSTRACT

We introduce the concept of *maximum mutual information decoding* for multiple-access channels. We then give conditions under which this *universal* decoding procedure can be effective on an arbitrarily varying multiple-access channel.

I. INTRODUCTION

In this paper we give conditions under which a simple decoding procedure, called the *maximum mutual information (MMI) decoding rule*, can be used to achieve certain information-transmission rate pairs over an arbitrarily varying multiple-access channel. The important feature of the MMI decoder is that it is *universal* in the sense that the receiver does not need to know the channel transition probability in order to implement the decoder.

An arbitrarily varying multiple-access channel is a transition probability W from $\mathcal{X}\times\mathcal{Y}\times\mathcal{S}$ into $\mathcal{Z}$, where $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{S}$, and $\mathcal{Z}$ are finite sets. We interpret $W(z|x,y,s)$ as the conditional probability of observing the channel output $z \in \mathcal{Z}$ given that the channel inputs were $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, and given that the channel *state* was $s \in \mathcal{S}$. For $\mathbf{x} = (x_1, \ldots, x_n) \in \mathcal{X}^n$, $\mathbf{y} \in \mathcal{Y}^n$, $\mathbf{s} \in \mathcal{S}^n$, and $\mathbf{z} \in \mathcal{Z}^n$, we set $W^n(\mathbf{z}|\mathbf{x},\mathbf{y},\mathbf{s}) \triangleq \prod_{k=1}^{n} W(z_k|x_k,y_k,s_k)$.

Definition. Let N, M, and n be positive integers. If f, g, and φ are mappings with

$$f : \{1,\ldots,N\} \to \mathcal{X}^n \quad \text{and} \quad g : \{1,\ldots,M\} \to \mathcal{Y}^n,$$

and

$$\varphi : \mathcal{Z}^n \to \{1,\ldots,N\} \times \{1,\ldots,M\},$$

then the triple (f, g, φ) is called a *code*. We call f an *encoder for user* 1. We call g an *encoder for user* 2, and we call φ a *decoder*. The *rate pair* of this code is the pair of nonnegative real numbers $(\frac{\log N}{n}, \frac{\log M}{n})$. Setting $\mathbf{x}_i \triangleq f(i)$, $i = 1, \ldots, N$, and $\mathbf{y}_j \triangleq g(j)$, $j = 1, \ldots, M$, we call $\mathbf{x}_1, \ldots, \mathbf{x}_N$ *codewords for user* 1, and we call $\mathbf{y}_1, \ldots, \mathbf{y}_M$ *codewords for user* 2. There is no requirement that the codewords be distinct. Clearly, knowing f and g is equivalent to knowing the codewords of each user.

Before discussing the MMI decoding rule, we need the following ideas. As described in [1], the *type* of an n-tuple $\mathbf{x} \in \mathcal{X}^n$ is defined to be the probability distribution $P_{\mathbf{x}}$ given by $P_{\mathbf{x}}(a) \triangleq N(a|\mathbf{x})/n$ for $a \in \mathcal{X}$, where $N(a|\mathbf{x})$ denotes the number of occurrences of a in the n-tuple $\mathbf{x}$. In an analogous way, we can define the *joint type* of a pair of n-tuples, $\mathbf{x}$ and $\mathbf{y}$, by setting $P_{\mathbf{x},\mathbf{y}}(a,b) \triangleq N(a,b|\mathbf{x},\mathbf{y})/n$ for $a \in \mathcal{X}$ and $b \in \mathcal{Y}$, where $N(a,b|\mathbf{x},\mathbf{y})$ denotes the number of occurrences of (a,b) in the n-tuple $((x_1,y_1),\ldots,(x_n,y_n))$.

In order to define the MMI decoding rule, we shall use the following notation. If $\mathbf{x} \in \mathcal{X}^n$, $\mathbf{y} \in \mathcal{Y}^n$, and $\mathbf{z} \in \mathcal{Z}^n$, then we set $I(\mathbf{y}\wedge\mathbf{z}|\mathbf{x}) \triangleq I(Y \wedge Z|X)$, where $P_{XYZ}(a,b,c) = P_{\mathbf{x},\mathbf{y},\mathbf{z}}(a,b,c)$ for $a \in \mathcal{X}$, $b \in \mathcal{Y}$, and $c \in \mathcal{Z}$. Other quantities such as $I(\mathbf{x} \wedge \mathbf{z})$ are defined similarly.

---

[1] This research was conducted while the author was with the Electrical Engineering Department and Systems Research Center, University of Maryland, College Park, MD 20742, U.S.A.

II. THE MAXIMUM MUTUAL INFORMATION DECODER

Consider the *decoding sets*

$$\hat{F}_i \triangleq \{\mathbf{z} \in \mathcal{Z}^n : I(\mathbf{x}_i \wedge \mathbf{z}) > I(\mathbf{x}_{i'} \wedge \mathbf{z}), \forall i' \neq i\}$$

and

$$\hat{G}_{ij} \triangleq \{\mathbf{z} \in \mathcal{Z}^n : I(\mathbf{y}_j \wedge \mathbf{z}|\mathbf{x}_i) > I(\mathbf{y}_{j'} \wedge \mathbf{z}|\mathbf{x}_i), \forall j' \neq j\}.$$

Obviously, the $\{\hat{F}_i\}_{i=1}^N$ are disjoint, as are the $\{\hat{G}_{ij}\}_{j=1}^M$ for each i. We say that $\hat{\varphi}(\mathbf{z}) = (\hat{\varphi}_1(\mathbf{z}), \hat{\varphi}_2(\mathbf{z}))$ is an MMI decoder if

$$\mathbf{z} \in \hat{F}_i \implies \hat{\varphi}_1(\mathbf{z}) = i \quad \text{and} \quad \mathbf{z} \in \hat{F}_i \cap \hat{G}_{ij} \implies \hat{\varphi}_2(\mathbf{z}) = j.$$

In other words, we first try to decode message i from user 1, and only then do we try to decode message j from user 2. Also, observe that since $\hat{\varphi}$ is defined in terms of the sets $\hat{F}_i$ and $\hat{G}_{ij}$, and since these sets in no way depend on W, $\hat{\varphi}$ is a universal decoder.

Definition. A pair of nonnegative real numbers, (R_1, R_2), is said to be *achievable* for W if for every $0 < \varepsilon < 1$, and every $\delta > 0$, for all sufficiently large n, there exist positive integers N and M such that $\frac{\log N}{n} > R_1 - \delta$ and $\frac{\log M}{n} > R_2 - \delta$, and such that there exists a code (f, g, φ) with

$$\frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} W^n(\{\mathbf{z} \in \mathcal{Z}^n : \varphi(\mathbf{z}) \neq (i,j)\}|\mathbf{x}_i, \mathbf{y}_j, \mathbf{s}) \ \leq \varepsilon, \qquad \forall \mathbf{s} \in \mathcal{S}^n.$$

Definition. If the decoder φ in the preceding definition can always be taken to be an MMI decoder, then we say that (R_1, R_2) is *achievable using MMI decoders.*

In order to state conditions under which an MMI decoder is useful, we need the following notation. Let P, Q, and T denote arbitrary probability distributions on $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{S}$, respectively. For the remainder of this paper, it shall be understood that

$$P_{XYSZ}(x, y, s, z) \triangleq P(x)Q(y)T(s)W(z|x, y, s).$$

As a consequence of this convention, expressions such as $I(X \wedge Z)$ and $I(Y \wedge Z|X)$ are now understood to depend implicitly on P, Q, T, and W. We further define

$$I^*(X \wedge Z) \triangleq \inf_T I(X \wedge Z) \quad \text{and} \quad I^*(Y \wedge Z|X) \triangleq \inf_T I(Y \wedge Z|X).$$

We now have the following result proved in [2].

Theorem. *Let P and Q be fixed. If for every distribution T on $\mathcal{S}$, we have*

$$I(X \wedge Z) > I(S \wedge Z) \quad \text{and} \quad I(Y \wedge Z|X) > I(S \wedge Z|X),$$

then every pair (R_1, R_2) satisfying

$$0 < R_1 < I^*(X \wedge Z) \quad \text{and} \quad 0 < R_2 < I^*(Y \wedge Z|X)$$

is achievable **using MMI decoders.**

REFERENCES

1. I. Csiszár and J. Körner, *Information Theory: Coding Theorems for Discrete Memoryless Systems.* New York: Academic, 1981.
2. J. A. Gubner, "Deterministic codes for arbitrarily varying multiple-access channels," Ph.D. dissertation, Electrical Engineering Department, University of Maryland, College Park, August, 1988.

BINARY CONTROL SYSTEMS DESIGN

R. T. YANUSHEVSKY

University of the District of Columbia
University of Maryland

ABSTRACT

The concept of binary system is considered. The class of binary systems is characterized by the presence of two channels of control actions. The first channel is a stabilizing one and it increases system reliability. The second channel ensures high efficiency of the closed-loop system. The procedure of the binary system design is considered. A link between binary and variable structure systems is examined.

1. INTRODUCTION

The concept of binary system characterizes a class of dynamic systems affine in control. It consists of two channels of control actions: the first channel makes the system asymptotically stable with satisfactory dynamics, the second serves to guarantee high efficiency of the system.

The presence of the second channel distinguishes binary systems from the wide-spread ordinary control systems [1, 2]. Its main feature is that it works the most effectively when the system coordinates do not belong to a prescribed domain and makes them move in this domain.

The contradiction between accuracy and stability in closed-loop control systems is well-known. The desire to increase gains in feedback systems is always connected with necessity to use special means to make the system asymptotically stable [2]. Even if such a goal is achieved the designed system becomes, as a rule, susceptible to noise. Moreover, large gains also cause saturation of system units. So the real system behavior will be different from the expected one.

A method has been proposed for building the so-called systems with variable structure which allow to reach high accuracy without using large gains [3-5]. One of the main features of these systems is the sliding mode which renders the system's dynamic characteristics of high quality. But a sliding mode, in itself, is a pure mathematical abstraction and does not exist in reality. Sliding modes are realized approximately in practice by high frequency oscillations. The so-called chattering decreases the quality of variable structure systems. To get rid of chattering control systems with sliding modes of high orders were considered which were called the binary systems [6, 7]. The given above concept of binary systems differs from that proposed in [6] and includes a wider class of control systems than examined in [3-7].

The method proposed here for binary system design is based on the optimal control theory approach, in particular, on the solution of special optimal control problems. At first, an optimal control problem is considered for an auxiliary dynamical system controlled by one part of control actions characterizing the first channel of control. Its goal is to make the system asymptotically stable with a satisfactory transient response but without using large gains. The most appropriate technique is based on the analytical controller design, and the parameters of the channel depend on coefficients of the integrand of the optimized functional [1, 2]. The solution of this subproblem is used to design the second channel of control. The analytical design problem of this channel is also formulated as an optimal control problem under the functional of a special type.

In a case of asymptotically stable original systems the first stage of the design in some cases may be omitted. For this reason the main attention will be paid to the second part of the formulated problem.

It should be mentioned that the presence of the first stabilizing channel is useful even in a case of stable systems because it increases their reliability.

For simplicity, but without loss of generality, we consider, at first, the design procedure for linear systems with a scalar control. Then we examine a nonlinear problem and also mentione a case of vector controls.

2. PROBLEM STATEMENT AND DESIGN PROCEDURE

Let us consider a controllable linear dynamic time-invariant system

$$\dot{x}(t) = Ax(t) + bu(t) \tag{2.1}$$

where $x(t)$ is an m-dimensional state vector, $u(t)$ is a control, A and b are a matrix and vector of appropriate dimensions.

Let the control units consists of two parts

$$u(t) = \nu_1(t) + \nu_2(t) \tag{2.2}$$

We can choose $\nu_1(t)$ following the well-known procedure minimizing

$$\min_{\nu_1} I = \int_0^{\infty} (x^T Qx + c\nu_1^T \nu)dt = V(x(0)) = x^T(0)Wx(0) \tag{2.3}$$

subject to (2.1) and (2.2), supposing $\nu_2(t) = 0$ [1, 2].
Then [1, 2]

$$\nu_1(t) = -\frac{1}{c} b^T Wx(t) \tag{2.4}$$

$$u(t) = \nu_2(t) - \frac{1}{c} b^T Wx(t) = \nu_2(t) - \sum_{i=1}^{m} k_i x_i(t) \tag{2.5}$$

where Q is a positive definite, or semidefinite, matrix; c, k_i are constant coefficients and the matrix $W = [w_{i_j}]$ satisfies the algebraic Riccati equation, or

$$\frac{dV}{dt}\Big|_{\nu_1(t)} = -x^T(Q + c^{-1}Wbb^TW)x \tag{2.6}$$

Let

$$\nu_2(t) = \sum_{i=1}^{r} z_i(t)x_i(t) \tag{2.7}$$

where $1 \leq r \leq m$ and $z_i(t), i = 1, \ldots, r$, are considered as new controls, which are bounded

$$|z| \leq M \tag{2.8}$$

($z(t) = \{z_1, \ldots, z_r\}$ and $M = \{M_1, \ldots, M_r\}, M_i$ are some constants).

The problem is how to determine such bounded controls $z_i(t)$ such that the system (2.1) will be of high efficiency.

Let

$$F(x) = x^T(Q + c^{-1}Wbb^TW)x + c_1 z_0^T z_0 \tag{2.9}$$

where $z_0(t) = \{z_{10}, \ldots, z_{r0}\}$ has the form

$$z_0(t) = -c_1^{-1}B^T(x)Wx \tag{2.10}$$

and c_1 is a constant,

$$B(x) = [bx_1 \ldots bx_r] \tag{2.11}$$

The proposed procedure is based on the following theorem.

Theorem 2.1. Given a positive definite function $V(x)$ (2.3) for the system (2.1) optimized by the control $\nu_1(t)$, then

(a) in some domain X_0 this function is the optimal Lyapunov function for the system (2.1) and (2.5) optimized by the control (2.7) with respect to the functional

$$I = \int_0^\infty (F(x) + c_1 z^T z)dt \tag{2.12}$$

(b) the system (2.1), (2.5), (2.7), (2.8) and (2.10) is asymptotically stable.

Using the dynamic programming procedure we can write Bellman's equation in the form

$$\min_z \left\{ F(x) + c_1 z^T z + \frac{\partial V^T}{\partial x}(Ax - c^{-1}bb^TWx) + \frac{\partial V^T}{\partial x}B(x)z \right\} = 0 \tag{2.13}$$

Assuming, at first, that the control $z(t)$ is unbounded, we can formally write the optimal solution

$$z(t) = -\frac{1}{2c_1}\, B^T(x)\, \frac{\partial V}{\partial x} = -c_1^{-1}B^T(x)Wx \tag{2.14}$$

and the equation (2.13) takes the form

$$F(x) - c_1 z^T z + \frac{\partial V^T}{\partial x}(Ax - c^{-1}bb^TWx) = 0 \tag{2.15}$$

But according to (2.3) and (2.6)

$$\frac{\partial V^T}{\partial x}(Ax - c^{-1}bb^T Wx) = -x^T(Q + c^{-1}Wbb^T W)x$$

and from (2.9) we can conclude that the equation (2.13) ((2.15)) is satisfied. Hence the system (2.1), (2.5) and (2.10) is optimal.

The inequalities (2.8) under controls (2.10) describe a domain X with a boundary ∂X and $\min_{\partial X} V(x) = \alpha$. Then the domain $\|x\| \leq \beta$ where $\max_{\|x\| \leq \beta} V(x) \leq \alpha$ will be X_0.

For the bounded control $z(t)$ we consider formally

$$B(x)z = B_1(x)z^* \tag{2.16}$$

where

$$z^*(t) = -c_1^{-1}B_1^T(x)Wx \tag{2.17}$$

$$B_1(x) = B(x)q \tag{2.18}$$

and $q = [q_{ii}]$ has the form

$$q_{ii} = \begin{cases} 1 & |z_i| \leq M_i \\ \frac{M_i}{z_{i0}} & |z_i| > M_i \end{cases}$$

The derivative of $V(x)$ along (2.1) under controls (2.5), (2.7) and (2.8) is given by

$$\begin{aligned} \frac{dV}{dt} &= \frac{\partial V^T}{\partial x}(Ax - c^{-1}bb^T Wx) + \frac{\partial V^T}{\partial x}B_1(x)z^* = \\ &= -x^T(Q + c^{-1}Wbb^T W)x - 2c_1 z^{*T} z \end{aligned}$$

so the system (2.1), (2.5), (2.7), and (2.8) is asymptotically stable.

3. SOME PROPERTIES OF BINARY SYSTEMS

From the last expression it follows that by using the additional specific channel of control actions we can significantly improve the efficiency of the closed-loop system.

Corollary 2.1. If a binary system quality is characterized by modules of the Liapunov function derivative along the system dynamical equation, then

$$|\frac{dV}{dt}|_u > |\frac{dV}{dt}|_{\nu_1} \tag{3.1}$$

where the compared derivatives is considered under the control $u(t)$ (2.5) and its part $\nu_1(t)$ (2.4).

It was mentioned that the inequalities (2.8) characterize the domain X of the "unbounded" controls which can be described as

$$|B(x)Wx| \leq c_1 M \tag{3.2}$$

By an appropriate choice of c_1 the domain X can be made "very thin" consisting of some hypersurfaces and attached layers of "thickness" $\varepsilon = c_1 M$. The control law makes the system coordinates move inside this domain.

When $c_1 \to 0$ this domain degenerates into hyposurfaces which describe the so-called sliding modes of systems with a variable structure [3-5].

The proposed approach to synthesis of binary systems allows to design systems with quality close to the well-known one of variable structure systems [3, 4] while at the same time doing away with chattering which restricted substantially the utilization of variable structure systems.

It should be mentioned that the controls $\nu_1(t)$ and $\nu_2(t)$ can be united and the control $u(t)$ can be presented in the form

$$u(t) = \sum_{i=1}^{m} \Psi_i(t) x_i(t) \tag{3.3}$$

where

$$\Psi_i = \begin{cases} M_i - k_i & z_i > M_i \quad (i = 1, \dots, r) \\ -c_1^{-1} x_i \sum_{s=1}^{m} \sum_{j=1}^{m} b_s w_{sj} x_j - k_i, & |z_i| \leq M_i \quad (i = 1, \dots, r) \\ -M_i - k_i & z_i < -M_i \quad (i = 1, \dots, r) \\ -k_i & \qquad (i = r+1, \dots, m) \end{cases} \tag{3.4}$$

Finally, in case of a vector control $u(t)$ we may choose some controls to stabilize, if it is necessary and possible, the control system or to improve its dynamics and others may be used analogous to $\nu_2(t)$. Of course, for the control vector $u(t)$ we have $z(t) = [z_{ij}]$ and instead of (3.4) we should consider a matrix $\Psi = [\Psi_{ij}]$.

4. NONLINEAR PROBLEM

Here we consider a class of nonlinear systems affine in control which can be stabilized by a control $\nu_1(t)$ (or they are asymptotically stable), and examine the channel of control $\nu_2(t) = x^T(t) z(t)$ (in any concrete case we can use the control (2.7)).

Let a nonlinear system under a control $\nu_1(t)$ have the following form with respect to $z(t)$ as a new control action

$$\dot{x}(t) = A(x,t) + B(x,t) z(t) \tag{4.1}$$

$$|z(t)| \leq M \tag{4.2}$$

where $A(x,t)$ and $B(x,t)$ are such matrix functions that the equation (4.1) has a unique continuous solution in a closed convex domain X_m of R_m; $A(0,t) = 0$.

Suppose also that we know the derivative of the Lyapunov function along (4.1) when $z(t) = 0$, i.e.

$$\dot{V}(x,t) = -Q(x,t) \tag{4.3}$$

$$F(x,t) = Q(x,t) + c_1 z_0^T(t) z_0(t) \tag{4.4}$$

where

$$z_0(t) = -\frac{1}{2c_1} B^T(x,t) \frac{\partial V(x,t)}{\partial x} \tag{4.5}$$

and $Q(x,t)$ is a positive definite function.

Theorem 4.1. Given a positive-definite function $V(x,t), x \in X_m$, with an infinitely small upper bound for the system (4.1), then

(a) in some domain X_0 this function is the optimal Lyapunov function for the system (4.1) and (4.2) optimized by the control (4.5) with respect to the functional

$$I = \int_{t_0}^{\infty} (F(x,t) + c_1 z^T z) dt \tag{4.6}$$

(b) the system (4.1), (4.2) and (4.5) is asymptotically stable in X_m.

The proof of this theorem is analogous to the proof of Theorem 2.1. We mention only that the inequalities (4.2) describe a domain $X(t)$ with the boundary $\partial X(t)$ and that for the system (4.1) and (4.2) the domain $X_0(\|x\| \leq \beta)$ can be written as

$$\sup_{\|x\| \leq \beta} V(x,t) \leq \inf_{x \in \partial X} V(x,t), \quad t > t_0$$

where ∂X is the boundary of the intersection of the domain $X(t)$ for $t \geq t_0$.

The domain $X(t)$ can be presented in the form

$$|B^T(x,t) \frac{\partial V(x,t)}{\partial x}| \leq 2c_1 M \tag{4.7}$$

For a small c_1 it can be made of a rather narrow form so the trajectories of the system (4.1) will approach to zero moving inside this prescribed domain.

5. EXAMPLE

Consider the procedure of building a binary structure for a system

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = -a_2 x_2 - a_1 x_1 + bu, \quad a_1 < 0;\ a_2 > 0; b > 0.$$

Let $\nu_1 = -k_1 x_1$ and $bk_1 > a_1$ so the closed-loop subsystem formed by the first channel of the control $\nu_1(t)$ is asymptomically stable.

Let $\nu_2 = zx_1$ and $|z| \leq M$. Following (2.9) and (2.10) we have

$$F(x) = 2\left\{w_{12}(a_1 + bk_1)x_1^2 + [w_{22}(a_1 + bk_1) + w_{12}a_2 - w_{11}]x_1x_2 + \right.$$
$$\left. + (w_{22}a_2 - w_{12})x_2^2\right\} + c_1z^2$$

and

$$z = -c_1^{-1}x_1(w_{12}x_1 + w_{22}x_2) \quad , |z| \leq M.$$

Here we chose the stabilizing control $\nu_1(t)$ without considering the optimal problem (2.3) and then determined the derivative $\dot{V}(t)$. Elements w_{ij} should guarantee the positive definiteness of the expression in braces.

From (3.4) follows

$$\Psi = \begin{cases} M - k_1 & x_1s > c_1M \\ -c_1^{-1}x_1s - k_1 & -c_1M \leq x_1s \leq c_1M \\ -M - k_1 & x_1s < -c_1M \end{cases}$$

where $s = w_{12}x_1 + w_{22}x_2$.

For $c_1M \to 0$ we immediately obtain the system with sliding hyperplane $s = 0$ [3, 4]. It is important also to emphasize that we used here only the output coordinate $x_1(t)$.

6. CONCLUSIONS

The concept of binary system was introduced as a system with two channels of control action. The first, stabilizing channel makes the system asymptotically stable with satisfactory dynamics, and the second channel guarantees high efficiency of the overall system. The stabilizing channel also increases the reliability of the system. An analytical design procedure for binary systems was proposed. A link between binary systems and variable structure systems based on sliding modes was established.

References

[1] T. Kailath, Linear Systems, (Englewood Cliffs, N.J.: Prentice-Hall, 1980).

[2] R. T. Yanushevsky, Theory of Linear Optimal Multivariable Control Systems, (Moscow, Nauka, 1973).

[3] V. I. Utkin, Sliding Modes and their Use in Variable Structure Systems, (Moscow, Nauka, 1974).

[4] U. Itkis, Control Systems of Variable Structure, (New York, Wiley, 1976).

[5] R. A. DeCarlo, S. H. Zakand and G. P. Mathew, "Variable Structure Control of Nonlinear Multivariable Systems", Proceeding of the IEEE, (March 1988), pp. 212-232.

[6] S. V. Yemel'yanov, S. K. Korovin, "Broadening of the set of types of feedbacks and application of them in the construction of Closed dynamic systems", Engineering Cybernetics, Vol. 19, No. 5, (1981), pp. 85-92.

[7] S. V. Yeml'yanov, S. K. Korovin and L. V. Levantovskii, "Second Order Sliding Modes in Controlling Uncertain Systems", Engineering Cybernetics, Vol. 24, No. 4, (1986), pp. 69-74.

ANALYSIS AND SYNTHESIS OF A CLASS OF NEURAL NETWORKS: VARIABLE STRUCTURE SYSTEMS WITH INFINITE GAINS

Jian-hua Li* Anthony N. Michel* Wolfgang Porod**
Department of Electrical and Computer Engineering
University of Notre Dame
Notre Dame, Indiana 46556

ABSTRACT

In the present paper we investigate the qualitative properties of a class of neural networks described by a system of first order ordinary differential equations with discontinuous right-hand side. We also develop an efficient synthesis procedure (resp., design procedure) for this class of neural networks.

The class of systems considered herein may be used as a representation of the analog Hopfield model with the nonlinearities having infinite gain. Also, under appropriate assumptions, the output of the class of systems considered herein may be viewed as representing the behavior of the discrete Hopfield model. Thus, our results give insight into the qualitative behavior of the analog as well as the discrete Hopfield models and they provide a means of designing such models.

The applicability of the present results is demonstrated by means of several specific examples.

I. INTRODUCTION

We will consider neural networks described by discontinuous differential equations of the form

$$dx/dt = F(x) \qquad (N)$$

with $F(x) = Ax+TS(x)+I$.We will define all symbols in (N) at a later point.

With an appropriate set of assumptions, which we will give later, this class of neural networks corresponds to the analog Hopfield model [2] with the gain of the non-linear functions infinite.

The structure of the remainder of this paper is as follows. In Section 2, we establish the notation used throughout. In Section 3, we present the system which we study and we compare it with both the analog and discrete Hopfield models. In Section 4, we state and prove the results summarized above under the category Analysis while in Section 5, we establish the results and procedure summarized above under the category Synthesis. The applicability of these results is demonstrated by means of specific examples in Section 6.

II. NOTATION

Let V and W be arbitrary sets. Then $V \cup W$, $V \cap W$, $V-W$ and $V \times W$ denote the union, intersection, difference and Cartesian product of V and W , respectively. If V is a subset of W , we write $V \subset W$ and if x is an element of V, we write $x \in V$. If f is a function from V into W , we write $f: V \to W$ and we let $f(U)=\{f(x) \in W : x \in U\}$ for $U \subset V$, and $f^{-1}(y) = \{x \in V : f(x)=y\}$ for $y \in W$. Let $\varnothing$ denote the empty set, let $\mathbf{R}$ denote the set of real numbers, and let $\mathbf{R}^{+}=[0,+\infty)$. If $V_1,\dots,V_n$ are n arbitrary sets, their Cartesian product is denoted by $\prod^n_{i=1} V_i = V_1 \times \cdots \times V_n$. If in particular, $V=V_1=\cdots=V_n$, we write $\prod^n_{i=1} V_i = V^n$. Let $\mathbf{R}^n$ be real n-space. If $x \in \mathbf{R}^n$, then $x^T=(x_1,\dots,x_n)$ denotes the transpose of x. When using a norm for $x \in \mathbf{R}^n$, $|x|$, we will have in mind $|x|=\sqrt{\Sigma^n_{i=1} x_i^2}$. If $x \in \mathbf{R}^n$ and $Y \subset \mathbf{R}^n$, then $x \perp Y$ will mean that $x^T \cdot y=0$ for all $y \in Y$. If $V \subset \mathbf{R}^n$, then $\overline{V}$, V^0 and ∂V represent the closure, interior and the boundary of V in $\mathbf{R}^n$, respectively. Let $\mathbf{B}^n = \{x \in \mathbf{R}^n : x_i=1 \text{ or } -1, i=1,\dots,n\}$. If

---

*The work of J.Li and A.N.Michel was supported by the National Science Foundation under grant ECS 88-02924.

**The work of W.Porod was supported by SDIO/IST and managed by the ONR under grant N00014-86-k-0506.

$[a_{ij}]$ is an arbitrary matrix, then A^T denotes the transpose of A. If A is a symmetric matrix, by $A>0$ we mean that A is positive definite and by $A\geq 0$ we mean that A is positive semidefinite. Sym(n) denotes the symmetric group of order n. If $\{x_1,\ldots,x_m\}\subset\mathbf{R}^n$, then $\mathrm{Span}(x_1,\ldots,x_m)$ denotes the subspace of $\mathbf{R}^n$ generated by $x_1,\ldots,x_m$ and $\mathrm{Aspan}(x_1,\ldots,x_m)$ denotes the affine subspace of $\mathbf{R}^n$ generated by $x_1,\ldots,x_m$. If $x_0\in\mathbf{R}^n$ and L is a subspace of $\mathbf{R}^n$, then $L+x_0$ denotes the affine subspace of $\mathbf{R}^n$ produced by shifting L by x_0, that is, $L+x_0 = \{y\in\mathbf{R}^n : y=x+x_0, x\in L\}$.

III. NEURAL NETWORK MODELS

We will consider neural networks described by discontinuous differential equations of the form

$$dx/dt = F(x) \tag{N}$$

with

$$F(x) = Ax+TS(x)+I \tag{3.1}$$

where $x=(x_1,\ldots,x_n)^T\in\mathbf{R}^n$, $F: \mathbf{R}^n\rightarrow\mathbf{R}^n$ is a measurable function, $A=\mathrm{diag}(-\rho_1,\ldots,-\rho_n)$ is an $n\times n$ constant matrix with each $\rho_i>0$, $T=[T_{ij}]$ is an $n\times n$ constant matrix, $I=(I_1,\ldots,I_n)^T$ is a constant vector and $S: \mathbf{R}^n\rightarrow\mathbf{R}^n$ is a measurable function defined by $S(x) = (\mathrm{sgn}(x_1),\ldots,\mathrm{sgn}(x_n))^T$.

Remark 3.1. 1) In [2] and [3], Hopfield considers a neural system which is represented by the system of equations

$$C_i(du_i/dt) = \sum_{j=0}^{n} T_{ij}v_j-u_i/R_i+I_i \quad , \quad i=1,\ldots,n \tag{H}$$

where $v_i=g_i(\lambda u_i)=(2/\pi)\tan^{-1}((\pi/2)\lambda u_i)$, $C_i>0$, $R_i>0$, $T_{ij}\in\mathbf{R}$, and $I_i\in\mathbf{R}$. This system can be realized by electronic circuits in which each nonlinear function g_i is implemented by a nonlinear (operational) amplifier with u_i the input voltage, v_i the output voltage and λ the gain of the amplifier. In practice, λ is a very large positive number. We want to know what will happen if λ becomes arbitrarily large. In this case, the nonlinear amplifiers may be viewed as hard threshold switches. To understand the behavior of this kind of circuit, we need to study an ideal mathematical model described by the set of equations

$$C_i(du_i/dt) = \sum_{j=0}^{n} T_{ij}v_j-u_i/R_i+I_i \quad , \quad i=1,\ldots,n \tag{H_n}$$

where $v_i=g_i(u_i)=\mathrm{sgn}(u_i)$. In ($H_n$), we may assume without loss of generality that $C_i=1$. If we let $(u_1,\ldots,u_n)^T= x$, $[T_{ij}]=T$, $\mathrm{diag}(-1/R_1,\ldots,-1/R_n)=\mathrm{diag}(-\rho_1,\ldots,-\rho_n)=A$, and $(I_1,\ldots,I_n)^T=I$, then system (H_n) assumes the form of system (N). Thus our model (N) is a modified analog Hopfield model (H) in which the gain of the nonlinear functions is high enough to be considered infinite.

2) Under certain conditions (which will be introduced later), the behavior of the output values of g_i agree with the discrete asynchronous model introduced by Hopfield in [1] as follows:

$$v_i^+ = \begin{cases} = \ 1 & \text{if} \sum_{j=1}^{n} T_{ij}v_j+I_i \ >0 \\ = \ -1 & \text{if} \sum_{j=1}^{n} T_{ij}v_j+I_i \ <0 \end{cases} \tag{N_d}$$

where $v=(v_1,\ldots,v_n)^T\in\mathbf{R}^n$ with $v_i=\pm1$, $T=[T_{ij}]$ is an $n\times n$ symmetric matrix with $T_{ii}=0$, and $I=(I_1,\ldots,I_n)^T$ is a constant vector. This model updates in an asynchronous manner in the sense that during each cycle, only one component of v is updated by the formula given above. ∎

IV. ANALYSIS

For each i, i=1,...,n, let

$$\mathbf{M}_i = \{ x=(x_1,\dots,x_n)^T \in \mathbf{R}^n : \ x_i = 0 \} \tag{4.1a}$$

and let

$$\mathbf{M} = \cup \mathbf{M}_i , \quad 1 \le i \le n. \tag{4.1b}$$

Then each $\mathbf{M}_i$ is a subspace of $\mathbf{R}^n$ of dimension (n-1). The right hand side function F(x) of the system (N) is defined only on $\mathbf{R}^n$-$\mathbf{M}$ and $\mathbf{M}$ is said to be the set of discontinuous surfaces of system (N). Note that $\mathbf{M}$ separates $\mathbf{R}^n$ into 2^n non-intersecting regions defined by

$$\mathbf{H}(\xi) = \mathbf{H}(\xi_1,\dots,\xi_n) = \{x \in \mathbf{R}^n : \ \xi_i x_i > 0 , \ 1 \le i \le n \} , \quad \xi \in \mathbf{B}^n . \tag{4.2}$$

On each $\mathbf{H}(\xi)$, the right hand side function F(x) is well defined and continuous. Thus the properties of the system (N), when confined to each $\mathbf{H}(\xi)$, can be studied in the usual way. On each $\mathbf{H}(\xi)$, the function $S(x) \equiv \xi$ is a constant and F(x) is a linear function which can be extended on $\mathbf{R}^n$ as

$$F_\xi : \mathbf{R}^n \to \mathbf{R}^n, \quad F_\xi(x) = Ax + (T\xi + I) = A(x - x_\xi) \tag{4.3a}$$

where

$$x_\xi = -A^{-1}(T\xi + I) . \tag{4.3b}$$

For each $\xi \in \mathbf{B}^n$, we define a linear differential equation as follows:

$$dx/dt = F_\xi(x) \tag{N$_\xi$}$$

where $x \in \mathbf{R}^n$. Since $F(x) \equiv F_\xi(x)$ on $\mathbf{H}(\xi)$, the behavior of system (N) is the same as the behavior of system (N_ξ) on each region $\mathbf{H}(\xi)$.

Since $A = \mathrm{diag}(-\rho_1,\dots,-\rho_n)$ is diagonal, the i-th component of (N_ξ) can be described as

$$dx_i/dt = F_{\xi i}(x) = -\rho_i(x_i - x_{\xi i}). \tag{N$_{\xi i}$}$$

Thus the components of a solution x=x(t) of (N_ξ) are independent of each other.

To study the properties of the system (N) on all of $\mathbf{R}^n$, we first have to give a proper definition of the solutions for (N). We require the notation given below. Let

$$\Lambda = \{ \xi = (\xi_1,\dots,\xi_n)^T : \quad \xi_i = \pm 1 \text{ or } 0 , 1 \le i \le n \}, \tag{4.4a}$$

$$\Lambda_m = \{ \xi = (\xi_1,\dots,\xi_n)^T \in \Lambda : \ \xi_{\sigma(i)} = 0, \ 1 \le i \le m \text{ and } \xi_{\sigma(i)} = \pm 1, \ m < i \le n , \ \sigma \in \mathrm{Sym}(n) \}, \ 0 \le m \le n. \tag{4.4b}$$

The number of elements in Λ_m is equal to $[n!/(m!(n-m)!)]2^{n-m}$, and the number of elements in $\Lambda = \cup \Lambda_m$, $0 \le m \le n$, is 3^n. For each $\xi \in \Lambda$, let

$$\mathbf{C}(\xi) = \{ \ x = (x_1,\dots,x_n)^T \in \mathbf{R}^n : \ x_i = 0 \text{ if } \xi_i = 0, \text{ and } x_i\xi_i > 0 \text{ if } \xi_i \ne 0, \}. \tag{4.4c}$$

Claim 4.1. For each m, $0 \le m \le n$, we can define by induction $\mathbf{S}_m$, $\mathbf{W}_m$ and $\mathbf{D}_m$ with the properties that $\mathbf{S}_m$, $\mathbf{W}_m$ are sets of functions from a subinterval of $[0,\infty)$ to $\mathbf{R}^n$, and $\mathbf{D}_m \subset \mathbf{R}^n$ such that

i) $\mathbf{D}_i \cap \mathbf{D}_m = \emptyset$ for $0 \le i < m$, (4.6a)

ii) $\mathbf{D}_0 \cup \cdots \cup \mathbf{D}_m \supset \mathbf{C}(\eta)$, for any $\mathbf{C}(\eta)$, $\eta \in \Lambda_m$, (4.6b)

and

iii) either $\mathbf{D}_m \supset \mathbf{C}(\gamma)$ or $\mathbf{D}_m \cap \mathbf{C}(\gamma) = \emptyset$ for any $\mathbf{C}(\gamma)$, $\gamma \in \Lambda_j$, $m < j \le n$. (4.6c)

Definition 4.1. A function φ in $\mathbf{S} = \cup \mathbf{S}_i$ is said to be a (local) solution of (N). A function $\phi = \phi(\cdot,\tilde{x})$ in $\mathbf{W} = \cup \mathbf{W}_i$ is said to be a (local) solution of (N) starting at $\tilde{x}$. In particular, solutions in $\mathbf{S}_0$ or $\mathbf{W}_0$ are said to be in the usual sense , while solutions in $\mathbf{S}_j$ or $\mathbf{W}_j$, j>0, are said to be in a sliding mode . ∎

Remark 4.1. 1) For each $\mathbf{H}(\xi)$, $\xi \in \mathbf{B}^n$, a function with its image in $\mathbf{H}(\xi)$ is a solution of system (N) in the sense of Definition 4.1 if and only if it is a solution of (N_ξ) in the usual sense.The definition of solutions of system (N) in the sliding mode with their images in the discontinuous surfaces $\mathbf{M}$ constitutes a generalized definition of solutions of discontinuous systems in the sense of Filippov(cf. [4] and [5]). ∎

Definition 4.2. 1) For $\tilde{x}\in \mathbf{R}^n$, a function $\phi=\phi(\cdot, \tilde{x})$: $[0,\tilde{t}) \to \mathbf{R}^n$ is said to be a solution of (N) starting at $\tilde{x}$ if a) $\phi(0, \tilde{x})=\tilde{x}$, and b) for any $t\in [0,\tilde{t})$, there is $\delta>0$ such that $t+\delta<\tilde{t}$ and ϕ restricted to $(t,t+\delta)$ is a (local) solution as defined in Definition 4.1.

2) The function ϕ is said to be in a sliding mode during the period $(t,t+\delta)$ if $\phi((t,t+\delta))\subset\mathbf{M}$. Also, ϕ is said to have no sliding mode if ϕ is not in a sliding mode on any interval. ▮

Theorem 4.1. 1) Every solution of system (N), $\phi=\phi(\cdot, \tilde{x})$: $[0, \tilde{t})\to\mathbf{R}^n$, can be extended to the interval $[0,\infty)$ in the sense that there is a solution of system (N), $\phi_1 =\phi_1(\cdot, \tilde{x})$: $[0,\infty)\to \mathbf{R}^n$, such that $\phi_1\equiv\phi$ on $[0, \tilde{t})$.

2) For any $\tilde{x}\in \mathbf{R}^n$, there is a solution of (N) starting at $\tilde{x}$ which is defined on the interval $[0,\infty)$.▮

In the remainder of this paper, we only consider solutions of (N) defined on the infinite interval $[0,\infty)$.

Definition 4.3. A vector $\tilde{x}\in \mathbf{R}^n$ is said to be an equilibrium (point) of (N) if the function $\phi(\cdot, \tilde{x})$: $[0,\infty)\to\mathbf{R}^n$ defined by $\phi(t,\tilde{x})\equiv\tilde{x}$, is a solution of (N). In particular, an equilibrium point of (N) in $\mathbf{R}^n$-$\mathbf{M}$ is said to be an equilibrium point in the usual sense. ▮

Definition 4.4. Let $\tilde{x}$ be an equilibrium of (N). 1) $\tilde{x}$ is said to be stable if for any $\varepsilon>0$, there is a $\delta=\delta(\varepsilon)>0$ such that $|\phi(t,0,x) - \tilde{x}|<\varepsilon$ for all $t\in [0,+\infty)$ whenever $|x-\tilde{x}|<\delta$. 2) $\tilde{x}$ is said to be asymptotically stable if (i) it is stable and (ii) there is an $\eta>0$ such that $\lim_{t\to+\infty}|\phi(t,0,x) - \tilde{x}|=0$ whenever $|x-\tilde{x}|<\eta$. 3) $\tilde{x}$ is said to be unstable if it is not stable. ▮

Theorem 4.2. For $\xi\in \Lambda_0$, consider $\mathbf{H}(\xi)$ and x_ξ as defined in (4.2) and (4.3). The following is true:

1) If $x_\xi\notin \mathbf{H}(\xi)$, there is no equilibrium point of (N) in the region $\mathbf{H}(\xi)$.

2) If $x_\xi\in \mathbf{H}(\xi)$, x_ξ is the unique equilibrium point of (N) in the region $\mathbf{H}(\xi)$, and x_ξ is also asymptotically stable and $\mathbf{H}(\xi)$ is contained in the domain of attraction of x_ξ.▮

Under the output of the system (N) we have in mind the variable $S(x)$ which resides in the set $\mathbf{B}^n$. For given $x\in \mathbf{R}^n$, we call $\xi=S(x)$ an output vector of system (N).

Definition 4.5. $\xi\in \mathbf{B}^n$ is said to be a stable output vector of system (N) if there is a solution $\phi=\phi(, \tilde{x})$: $[0,+\infty)\to\mathbf{R}^n$ of system (N) and $\tilde{t}\in [0, +\infty)$ such that $\phi((\tilde{t},+\infty))\subset\mathbf{H}(\xi)$. $\xi\in \mathbf{B}^n$ is said to be an unstable output vector of system (N) if ξ is not a stable output vector. ▮

Theorem 4.4. 1) For any $\xi\in \mathbf{B}^n$, ξ is a stable output vector of system (N) if and only if $x_\xi \in \overline{\mathbf{H}(\xi)}$.

2) As t increases , if the output vector of system (N) reaches a stable output vector ξ , then the corresponding solution of (N) will converge to the stable equilibrium point x_ξ as $t\to+\infty$.

3) If no component of x_ξ is 0, for any $\xi\in \mathbf{B}^n$, the map $\xi\to x_\xi$ determines a one to one correspondence between the set of stable output vectors of system (N) and the set of stable equilibrium points of system (N) in $\mathbf{R}^n$-$\mathbf{M}$. ▮

Assumption 4.1. For system (N), we will assume: 1) $T=[T_{ij}]$ is symmetric, 2) $T_{ii} \geq 0$, $1\leq i\leq n$, and 3) no component of x_ξ is 0 for any $\xi\in \mathbf{B}^n$.▮

Remark 4.4. For system (N), consider a fixed $n\times n$ matrix T. Then for almost all vectors $I\in \mathbf{R}^n$, Part 3 of the Assumption 4.1 is true.▮

Definition 4.6. If system (N) satisfies Assumption 4.1 then 1) the energy function E: $\mathbf{B}^n \to \mathbf{R}$ of system (N) is defined as

$$E(\xi) = -(1/2)\,\xi^T T \xi - \xi^T I + c, \tag{4.11}$$

where c is a constant, and 2) $\xi \in \mathbf{B}^n$ is said to be a (local) minimum of the energy function E if $E(\xi) \le E(\eta)$, for $\eta \in \mathbf{J}(\xi)$, where

$$\mathbf{J}(\xi) = \{\, \eta \in \mathbf{B}^n \ : \ \text{there is a k such that } \eta_k \xi_k < 0, \text{ and } \eta_i \xi_i > 0,\ i \neq k \}. \tag{4.12}$$

Also for any $\eta \in \mathbf{J}(\xi)$, η is said to be an adjacent vector of ξ in $\mathbf{B}^n$. ∎

Theorem 4.5. Assume that system (N) satisfies Assumption 4.1. Then the following is true: 1) A local minimum of the energy function E is a stable output vector of system (N).

2) If $T_{ii}=0$, $1 \le i \le n$, then a stable output vector of system (N) is a local minimum of the energy function E. ∎

Remark 4.5. 1) Theorem 4.5 shows that there is a one to one correspon-dence between the set of local minima of E and the set of stable output vectors of system (N) under the condition $T_{ii}=0$, $1 \le i \le n$. Without this condition, the output of system (N) may get stuck at some vector which is not a local minimum of the energy function E. Therefore, when using the energy function to synthesize system (N) , we will want T_{ii} as small as possible, provided that $T_{ii} \ge 0$.

2) When there is a negative T_{ii} , the energy function E for system (N), given in (4.11), need not necessarily decrease monotonically with increasing time. This is quite different from the behavior of the analog Hopfield model (c.f. [2] and [8]). ∎

Theorem 4.6. If system (N) satisfies Assumption 4.1, no solution of system (N) will be in the sliding mode during any time period, i.e., $\mathbf{D}_o = \mathbf{R}^n$. ∎

Definition 4.7. Assume that system (N) satisfies Assumption 4.1. According to Theorem 4.6, for any solution of system (N), $\varphi = \varphi(\ ,x_0) : [0,+\infty) \to \mathbf{R}^n$, we can define a constant k_φ and two associated sequences $\mathbf{T}_\varphi = \{t_i \in (0,+\infty) : 1 \le i \le k_\varphi\}$ and $\mathbf{B}_\varphi = \{\xi_i \in \mathbf{B}^n : 1 \le i \le k_\varphi\}$ by induction as follows: i) For m=1, by Theorem 4.6. , there is $\varepsilon>0$ and $\xi \in \mathbf{B}^n$ such that $\varphi((0,\varepsilon)) \subset \mathbf{H}(\xi)$. Let $\xi_1=\xi$ and let $t_1 = \sup\{\, t \in (0,+\infty) : \varphi((0,t)) \subset \mathbf{H}(\xi_1)\}$. If $t_1 = +\infty$, let $k_\varphi = 1$. ii) For m>1, suppose that t_i and ξ_i, $1 \le i \le m-1$, have been defined and $t_{m-1} \neq +\infty$. By Theorem 4.6. , there is $\varepsilon>0$ and $\xi \in \mathbf{B}^n$ such that $\varphi((t_{m-1}, t_{m-1}+\varepsilon)) \subset \mathbf{H}(\xi)$. Let $\xi_m=\xi$ and let $t_m = \sup\{\, t \in (t_{m-1},+\infty) : \varphi((t_{m-1},t)) \subset \mathbf{H}(\xi_m)\}$. If $t_m = +\infty$ or $m=2^n$, let $k_\varphi = m$. iii) If for any finite integer m, $t_m \neq +\infty$, let $k_\varphi = +\infty$. $\mathbf{B}_\varphi$ is said to be the output sequence of system (N) associated with the solution φ, k_φ is said to the length of $\mathbf{B}\varphi$, and $\mathbf{T}_\varphi$ is said to be the time sequence associated with solution φ. ∎

Theorem 4.7. Suppose that system (N) satisfies Assumption 4.1, and $\mathbf{T}_\varphi = \{t_i \in (0,+\infty) : 1 \le i \le k_\varphi\}$ and $\mathbf{B}_\varphi = \{\xi_i \in \mathbf{B}^n : 1 \le i \le k_\varphi\}$ are the two associated sequences of a solution φ of system (N). If $T \ge 0$, i.e., T is positive semidefinite, then 1) $E(\xi_{i+1}) < E(\xi_i)$, $1 \le i < k_\varphi$, 2) $k_\varphi \le 2^n$ and $t_{k\varphi} = +\infty$, and 3) $\xi_{k\varphi}$ is a stable output vector of (N). ∎

When T is not positive semidefinite, we may consider a special kind of solution of (N) for which the results of Theorem 4.7. still hold.

Definition 4.8. 1) For ξ , $\eta \in \mathbf{B}^n$, η is said to be reachable from ξ if there is a sequence $\{\beta_0, \ldots, \beta_m\} \subset \mathbf{B}^n$ such that $\xi=\beta_0$, $\eta=\beta_m$ and for each i, $1 \le i < m$, β_i and β_{i+1} are adjacent. Furthermore if β_i and β_{i+1} differ at the l_i-th component, then $\beta_{i,l_i} \cdot x_{\beta_i,l_i} < 0$, where $x_{\beta_i} = T\beta_i + I$, and β_{i,l_i} , x_{β_i,l_i} are the l_i-th component of β_i and x_{β_i} , respectively.

2) A solution $\varphi = \varphi(\ ,x_0) : [0,+\infty) \to \mathbf{R}^n$ of system (N) is said to be an asynchronous solution of system (N), if for each i, $1 < i \le k_\varphi$, ξ_{k+1} is reachable from ξ_k. ∎

Theorem 4.8. Suppose that system (N) satisfies Assumption 4.1. For any asynchronous solution φ of system (N) with the two associated sequences $\mathbf{T}_\varphi = \{t_i \in (0,+\infty) :$

$1 \leq i \leq k_\varphi$ } and $\mathbf{B}_\varphi = \{\xi_i \in \mathbf{B}^n : 1 \leq i \leq k_\varphi\}$, we have 1) $E(\xi_{i+1}) < E(\xi_i)$, $1 \leq i < k_\varphi$, 2) $k_\varphi \leq 2^n$ and $t_{k\varphi} = +\infty$, 3) $\xi_{k\varphi}$ is a stable output vector of (N), and 4) $\{\xi_i \in \mathbf{B}^n : 1 \leq i \leq k_\varphi\}$ is a subsequence of a solution of the asynchronous discrete system (N_d).▮

Remark 4.7. In Theorem 4.8, Part 1 states that the energy function decreases monotonically along the output sequence. Parts 2 and 3 state that the output sequences do not oscillate and they will converge to stable output vectors of (N) during a finite number of steps. Part 4 establishes a relationship between system (N) and the discrete Hopfield model, provided that T_{ii} may also be positive.▮

Definition 4.9. Suppose that system (N) satisfies Assumption 4.1. A directed graph $\mathbf{G}=(\mathbf{V}, \mathbf{A})$ associated with system (N) is defined as follows: 1) $\mathbf{V} = \mathbf{B}^n$, and 2) $\mathbf{A} = \{(\xi,\eta) \in \mathbf{V} \times \mathbf{V} :$ η is adjacent to ξ and if ξ and η differ in the k-th component then $x_{\xi k}\xi_k < 0$ where $x_\xi = T\xi + I$ }. The elements in $\mathbf{V}$ are the vertices of $\mathbf{G}$ and the elements in $\mathbf{A}$ are the arcs of $\mathbf{G}$ (cf. [12]). For $\xi, \eta \in \mathbf{V}$, η is said to be a descendant of ξ if there is a path from ξ to η. For $\xi \in \mathbf{V}$, ξ is said to be a leaf of graph $\mathbf{G}$ if ξ has no descendants other than itself.▮

Definition 4.10. Suppose that system (N) satisfies Assumption 4.1, and $\xi \in \mathbf{B}^n$ is a stable output vector of (N). $\eta \in \mathbf{B}^n$ is said to be in the domain of attraction of ξ if η is the only leaf of $\mathbf{G}$ which is a descendant of ξ, where $\mathbf{G}$ is the directed graph associated with system (N) as defined in Definition 4.9.▮

V. SYNTHESIS

By properly scaling each component of x, we may consider the case A= diag(-1,..., -1). In this case, the system assumes the form

$$dx/xt = -x + TS(x) + I\,. \tag{N}$$

If we take T= n×n identity matrix and I= 0, then for the system

$$dx/dt = -x + S(x)\,, \tag{N_{id}}$$

every $\alpha \in \mathbf{B}^n$ is a stable output vector. In view of the above discussion, the basic synthesis problem which we want to address may be stated as follows

Synthesis problem: Given m vectors in $\mathbf{B}^n$, say $\alpha_1, \ldots, \alpha_m$, with the matrix A= diag(-1, ...,-1) and the nonlinear function $S(x) = (\mathrm{sgn}(x_1), \ldots, \mathrm{sgn}(x_n))^T$ fixed, how can we properly choose an optimal pair {T, I} such that the designed system (N) has the properties enumerated below ?

1) $\alpha_1, \ldots, \alpha_m$ are stable output vectors of system (N).

2) The system has no oscillatory output sequences.

3) The total number of the spurious stable output vectors (i.e., stable output vectors of (N) contained in $\mathbf{B}^n - \{\alpha_1, \ldots, \alpha_m\}$) is as small as possible.

4) The domain of attraction of each α_i is as large as possible.

In view of the analysis results developed in the previous section, we may solve the preceding synthesis problem in the following manner.

Synthesis strategy: Given m vectors in $\mathbf{B}^n$, say $\alpha_1, \ldots, \alpha_m$, find an n×n matrix $T = [T_{ij}]$ and a vector $I = (I_1, \ldots, I_n)^T$ such that:

1) T is symmetric and $T_{ii} \geq 0$, i=1,...,n.

2) $x_{\alpha_i} = \alpha_i$, where $x_{\alpha_i} = T\alpha_i + I$. (5.1)

3) Each α_i is a (local) minimum of the energy function E of the synthesized system, where $E: \mathbf{B}^n \to \mathbf{R}$ is defined by

$$E(\alpha) = -(1/2)\,\alpha^T T \alpha - \alpha^T I\,, \tag{5.2}$$

as in Definition 4.5.

4) Each diagonal element of T is as small as possible.▮

Remark 5.2. The present synthesis problem was first addressed by Hopfield for the discrete model (N_d) in [1]. In [7], Personnaz gave another synthesis procedure. The above two methods will be compared with our method in the next section. ▮

Suppose that we are given m vectors in $\mathbf{B}^n$, say $\alpha_1, ..., \alpha_m$. Let

$$L = \mathrm{Span}(\alpha_1 - \alpha_m, ..., \alpha_{m-1} - \alpha_m) \tag{5.3a}$$

and

$$L_a = \mathrm{Aspan}(\alpha_1, ..., \alpha_m). \tag{5.3b}$$

Then L is the linear subspace of $\mathbf{R}^n$ generated by the (m-1) vectors $\alpha_1 - \alpha_m, ..., \alpha_{m-1} - \alpha_m$, L_a is the affine subspace of $\mathbf{R}^n$ generated by the vectors $\alpha_1, ..., \alpha_m$, and $L_a = L + \alpha_m$. Assume that $k = \mathrm{rank}(L)$, $\{u_1, ..., u_k\}$ is an orthonormal basis of L, and $\{u_{k+1}, ..., u_n\}$ is an orthonormal basis of $L_1^{\perp}$. Let $U^+ = [u_1, ..., u_k]$, $U^- = [u_{k+1}, ..., u_n]$,

$$T^+ = [T^+_{ij}] = \sum_{i=1}^{k} u_i u_i^T = U^+(U^+)^T, \quad T^- = [T^-_{ij}] = \sum_{i=k+1}^{n} u_i u_i^T = U^-(U^-)^T, \tag{5.4}$$

and let

$$T_\tau = T^+ - \tau T^-, \text{ and } I_\tau = \alpha_m - T_\tau \alpha_m, \tag{5.5}$$

where $\tau \in \mathbf{R}$ is a parameter. For this class of synthesized systems

$$dx/dt = -x + T_\tau S(x) + I_\tau \qquad \tau \in \mathbf{R} \tag{N_τ}$$

we have the following result.

Theorem 5.1. 1) For any $\tau \in \mathbf{R}$, each $\alpha \in \mathbf{B}^n \cap L_a$ is a stable output vector of system (N_τ). In particular, this is true for $\alpha_1, .., \alpha_m$.

2) For any $\tau \in \mathbf{R}$, T_τ is symmetric.

3) When $\tau > -1$, each $\alpha \in \mathbf{B}^n \cap L_a$ is a global minimum of the energy function E. In particular, this is true for $\alpha_1, .., \alpha_m$.

4) When $\tau > -1$, the value of each main diagonal element of T_τ and the number of spurious stable output vectors of system (N_τ) decrease as τ increases.

5) When $-1 < \tau \leq \tau_{op}$, where

$$\tau_{op} = \min\{ T^+_{ii} / T^-_{ii}, \ 1 \leq i \leq n \} \tag{5.6}$$

where $T^+_{ii} = \sum_{l=1}^{k} (u_{l,i})^2$, $T^-_{ii} = \sum_{l=k+1}^{n} (u_{l,i})^2$ are the (i,i)th elements of T^+ and T^-, respectively,

then each $\alpha \in \mathbf{B}^n \cap L_a$ behaves as an attractor. In particular, this is true for $\alpha_1, ..., \alpha_m$.

6) Taking $\tau = 0$, we obtain the system

$$dx/dt = -x + T_0 S(x) + I_0, \tag{N_0}$$

where

$$T_0 = T^+, \ I_0 = \alpha_m - T_0 \alpha_m. \tag{5.7}$$

The system (N_0) is optimal in the sense that it has the fewest number of spurious stable output vectors among the systems (N_τ), $\tau > -1$, having no oscillatory output sequences with respect to all of their solutions.

7) Taking $\tau = \tau_{op}$, we obtain the system

$$dx/dt = -x + T_{op} S(x) + I_{op}, \tag{N_{op}}$$

where

$$T_{op} = T^+ - \tau_{op} T^-, \ I_{op} = x_m - T_{op} x_m. \tag{5.8}$$

The system (N_{op}) is optimal in the sense that it has the fewest number of spurious stable output vectors among systems (N_τ), $\tau > -1$, having no oscillatory output sequences with respect to all of their asynchronous solutions.▮

To detect spurious stable output vectors, we make use of the following result.

Theorem 5.2. For system $(N\tau)$, $-1<\tau\leq\tau_{op}$, any of the spurious stable output vectors which do not belong to L_a can be detected by comparing the final output vector α_∞ and final state vector x_{α_∞}. If $\alpha_\infty \neq x_{\alpha_\infty}$, α_∞ is not in the given data set $\{\alpha_1,\dots,\alpha_m\}$.∎

Remark 5.4. 1) The synthesis method introduced here can be applied directly to the discrete Hopfield model (N_d) which is described in Section III.

2) In practice, we may synthesize two sets of $\{T_\tau, I_\tau\}$ for $\tau = \tau_{op}$ and $\tau = \tau'$, where τ' is a large positive number. With a given initial condition, we run system (N_{op}) for a sufficient time until the output is settled down. Then we use the output vector as an initial condition to run system $(N_{\tau'})$. By Theorem 5.1.4 and Lemma 5.3.1, most of the spurious stable output vectors of (N_{op}) in $\mathbf{B}^n$-L_a will turn out to be unstable output vectors of $(N_{\tau'})$ and hence will be detected. This will be illustrated in Examples 1 and 2 in the next section.

3) Any vector in $L_a \cap \mathbf{B}^n$ will always be stored in the system (N_τ) as a stable output vector even if it is not in the given desired set of vectors $\{\alpha_1,\dots,\alpha_m\}$ to be stored. This can sometimes be used to simplify our synthesis procedure as will be demonstrated in Example 3 in the next section.

4) When $\tau>\tau_{op}$, system (N_τ) may exhibit oscillations although it may have fewer spurious stable output vectors. However, with properly chosen large positive τ, T_τ and I_τ may be used to synthesize the analog Hopfield model. The resulting system will not oscillate and the set of stable equilibrium points in $\mathbf{B}^n$ is approximately equal to $\mathbf{B}^n \cap L_a$ (c.f. [9]).∎

VI. SYNTHESIS PROCEDURE AND EXAMPLES

Synthesis Procedure: Suppose that we are given m vectors $\alpha_1,\dots,\alpha_m$ in $\mathbf{B}^n$, which are to be stored as stable output vectors for an n dimensional system (N). We proceed as follows:

1) Compute the $n\times(m-1)$ matrix $Y=[\alpha_1-\alpha_m, \dots, \alpha_{m-1}-\alpha_m]$.

2) Perform a singular value decomposition of Y and obtain the matrices U, V and Σ such that $Y = U\Sigma V^T$, where U and V are unitary matrices and where Σ is a diagonal matrix with the singular values of Y on its diagonal. (This can be accomplished by standard computer routines.). Let $Y=[y_1,\dots,y_{m-1}]$, $U=[u_1,\dots, u_n]$, and k=dimension of $\mathrm{Span}(y_1,\dots, y_{m-1})$. From the properties of singular value decomposition, we know that k=rank of Σ, $\{u_1,\dots, u_k\}$ is an orthonormal basis of $\mathrm{Span}(y_1,\dots, y_{m-1})$ and $\{u_1,\dots, u_n\}$ is an orthonormal basis of $\mathbf{R}^n$.

3) Compute $T^+ = [T^+_{ij}] = \sum_{i=1}^{k} u_i u_i^T$, and $\quad T^- = [T^-_{ij}] = \sum_{i=k+1}^{n} u_i u_i^T$.

4) Choose a value for the parameter $\tau>-1$ and compute

$$T_\tau = T^+ - \tau T^-, \text{ and } I_\tau = \alpha_m - T_\tau\alpha_m.$$

In particular, the optimal pair $\{T_{op}, I_{op}\}$ is given by

$$T_{op} = T^+ - \tau_{op} T^-,\ I_{op} = \alpha_m - T_{op}\alpha_m, \text{ where } \tau_{op} = \min\{\ T^+_{ii}/T^-_{ii}\ ,\ 1\leq i\leq n\}.$$

Then all vectors in $L_a \cap \mathbf{B}^n$, where $L_a=\mathrm{Aspan}(\alpha_1,\dots,\alpha_m)$, including $\alpha_1,\dots,\alpha_m$, will be stored in the system

$$dx/dt = -x + T_\tau S(x) + I_\tau \qquad (N_\tau)$$

as stable output vectors of (N_τ).∎

Example 1. The dimension of the system to be considered is n=10. Given are m=5 vectors specified by

$$\alpha_1 = \begin{pmatrix} -1 \\ 1 \\ -1 \\ 1 \\ 1 \\ 1 \\ -1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \quad \alpha_2 = \begin{pmatrix} 1 \\ 1 \\ -1 \\ -1 \\ 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ 1 \end{pmatrix}, \quad \alpha_3 = \begin{pmatrix} -1 \\ 1 \\ 1 \\ 1 \\ -1 \\ -1 \\ 1 \\ -1 \\ 1 \\ -1 \end{pmatrix}, \quad \alpha_4 = \begin{pmatrix} 1 \\ 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \quad \alpha_5 = \begin{pmatrix} 1 \\ -1 \\ -1 \\ -1 \\ 1 \\ 1 \\ 1 \\ -1 \\ -1 \\ -1 \end{pmatrix}.$$

1) Compute the $n\times(m-1) = 10\times4$ matrix
$Y = [y_1, y_2, y_3, y_4]$, where $y_i = \alpha_i - \alpha_5$ for $1\leq i\leq 4$.

2) Perform a singular value decomposition of Y to obtain the matrices U, V and Σ such that $Y = U\Sigma V^T$. We have, k = rank of $\Sigma = 4$, and $U = [u_1,..., u_{10}]$.

3) Compute
$T^+ = [T^+_{ij}] = u_1u_1^T + ... + u_4u_4^T$, and $T^- = [T_{ij}] = u_5u_5^T + ... + u_{10}u_{10}^T$

4) Compute
$\tau_{op} = 0.33010$, $T_{op} = T^+ - \tau_{op} T^-$ and $I_{op} = \alpha_m - T_{op}\alpha_m$.
By Theorem 4.4, we determine that the synthesized system

$$dx/dt = Ax + T_{op} S(x) + I_{op} \qquad (N_{op})$$

has 20 stable output vectors including $\alpha_1,...,\alpha_5$. We slao have that $L_a \cap B^n = \{ \alpha_1,...,\alpha_m\}$.

In Table 1 below, the total number of stable output vectors of the synthesized system for different values of τ are listed and compared with the other two methods discussed previously.▮

Table 1

	Hopfield's Method	Personnaz' Method	Present Method			
	(cf. [1])	(cf.[7])	$\tau=0$	$\tau=\tau_{op}$	$\tau=1$	$\tau=10$
total number of stable output vectors	4	122	58	20	8	5
total number of errors	5	0	0	0	0	0
total number of adjacent vectors in the domains of attraction of α_i, $1\leq i\leq 5$	0	30	40	40	no test	no test

Example 2: We repeated Example 1 50 times with 50 different sets of given vectors to be stored as stable output vectors of system (N). Each set of vectors contains m=5 vectors. For each given set of vectors, we synthesized a system (N) as in Example 1. Table 2 displays the average of these results and compares them with other methods.▮

Table 2

	Hopfield's Method	Personnaz' Method	Present Method			
	(cf. [1])	(cf.[7])	$\tau=0$	$\tau=\tau_{op}$	$\tau=1$	$\tau=10$
average of total number of stable output vectors	5.52	108	49.2	30.3	8.50	7.40
average of total number of errors	3.70	0	0	0	0	0
average of total number of adjacent vectors in the domains of attraction of α_i, $1\leq i\leq 5$	3.16	28.2	37.1	37.9	no test	no test

Remark 6.2. By the results listed in Tables 1 and 2, we see that :

1) Using the present method, the $\alpha_1,\dots,\alpha_5$ are always stored as stable output vectors. Also, as τ increases, the number of spurious stable output vectors decreases. Furthermore, when $\tau \le \tau_{op}$, the extent of the domain of attraction of each α_i , $1 \le i \le 5$, increases as τ increases.

2) In view of these observations, the present synthesis method appears to offer improvements over some of the existing methods.∎

Example 3: (For a detailed analysis cf. [9].) Let us consider patterns made up of 10×10 small boxes. Each pattern corresponds to a vector in $\mathbf{R}^{100}$ with each component value varying form -1 to 1 determined by the gray level in the corresponding box. If the gray level in a box is white the value of the corresponding component is -1. If the gray level in a box is black, the value of the corresponding component is 1. Suppose that we want to synthesize a system (N) with n=100 neurons, and the vectors $v_{00},\dots,v_{99}$, which correspond to the patterns $\boxed{00},\dots,\ \boxed{99}$, respectively, are to be stored as stable output vectors of system (N). Instead of actually storing the m=100 vectors $v_{00},\dots,v_{99}$, we may consider to store only m=21 vectors $v_{0\phi},\dots,v_{9\phi}$, $v_{\phi 0},\dots,$ $v_{\phi 9}$ and v_ϕ, which correspond to the patterns $\boxed{0\ },\dots,$ $\boxed{9\ }$, $\boxed{\ 0},\dots,$ $\boxed{\ 9}$ and $\boxed{}$, respectively. Since $v_{ij}=(v_{i\phi}-v_\phi)+(v_{\phi j}-v_\phi)+v_\phi$ for $0 \le i,j \le 9$, $v_{00},\dots,v_{99}$ are contained in the affine subspace generated by $v_{0\phi},\dots,v_{9\phi}$, $v_{\phi 0},\dots,$ $v_{\phi 9}$ and v_ϕ. Thus, by Theorem 5.1, $v_{00},\dots,v_{99}$ will also be stored as stable output vectors of the synthesized system.∎

REFERENCES

[1] J.J. Hopfield, " Neural networks and physical systems with emergent collective computational abilities", Proc. Natl. Acad. Sci. USA, Biophysics, Vol. 79, pp2554-2558, April 1982.

[2] J.J. Hopfield, " Neurons with graded response have collective computational properties like those of two-state neurons", Proc. Natl. Acad. Sci. USA, Vol. 81, pp. 3088-3092, May 1984.

[3] D.W.Tank and J.J. Hopfield,"Simple 'neural' optimization networks: an A/D converter, signal decision circuit, and a linear programming circuit", IEEE Trans. on Circuits and Systems, Vol CAS-33, No. 5, pp. 533-541, May 1986.

[4] A.N.Michel and D.W.Porter,"Practical stability and finite-time stability of discontinuous systems", IEEE Trans. on Circuit Theory, Vol. CT-19, No. 2, pp. 123-129, March 1972.

[5] V.I. Utkin, " Variable structure systems with sliding modes", IEEE Trans. Auto. Control, Vol. AC-22, N0. 2, pp. 212-222, April 1977.

[6] R.K. Miller and A.N. Michel, Ordinary Differential Equations, Academic Press, New York,1982.

[7] L. Personnaz, I. Guyon, and G. Dreyfus, " Collective computational properties of neural networks: new learning mechanisms", Physical Review A, The American Physical Society, Vol. 34, Number 5, pp4217-4228, Nov. 1986.

[8] J. Li, A.N. Michel and W. Porod, " Qualitative analysis and synthesis of a class of neural networks", IEEE Trans. Circuits and Systems, Vol. 35, No. 8, pp. 976-986, Aug. 1988.

[9] J. Li, " Qualitative Analysis and Synthesis of a Class of Neural Networks ", Ph.D, Dissertation, University of Notre Dame, 1988.

[10] D.L. Luenberger, Optimization by Vector Space Methods, John Willy and Sons, New York 1969.

[11] A.N. Michel and C.J. Herget, Mathematical Foundations in Engineering and Science, Prentice-Hall, Englewood Cliffs, N.J. , 1981.

[12] A.V. Aho, J.E. Hopcroft and J.D. Ullman, Data Structures and Algorithms, Addison-Wesley Publishing Company, 1987.

ON THE ESTIMATION OF SLIDING DOMAINS AND STABILITY REGIONS OF VARIABLE STRUCTURE CONTROL SYSTEMS

S. MEHDI MADANI-ESFAHANI*
School of Electrical Engineering
Purdue University
West Lafayette, IN 47907

STEFEN HUI**
Department of Mathematical Sciences
San Diego State University
San Diego, CA 92182-0314

STANISLAW H. ŻAK*
School of Electrical Engineering
Purdue University
West Lafayette, IN 47907

ABSTRACT

This paper examines the problem of the control of multivariable linear time-invariant systems with bounded controllers using the variable structure control (VSC) approach. A coordinate transformation which decouples the fast and slow states is utilized, sliding domains and estimates of the domain of attraction along with the regions of asymptotic stability (RAS) are obtained with the aid of Lyapunov type arguments.

1. INTRODUCTION

Variable Structure Control (VSC), the control of dynamical systems with discontinuous feedback controllers, has been developed over the last 25 years. See [1], [2] and [6] for surveys. See also [10]. This theory rests on the concept of changing the structure of the controller in response to the changing states of the system to obtain a desired response. This is accomplished by the use of a high speed switching control law which forces the trajectories of the system onto a chosen manifold, where they are maintained thereafter. The system is insensitive to certain parameter variations and disturbances while the trajectories are on the manifold.

In this paper we consider linear time-invariant multivariable control systems. We use a transformation ([12]) for decoupling the "fast" and "slow" states. The fast states describe the motion of the system off the manifold and the slow states describe the motion on the manifold. The transformation permits a simpler controller design. Once the controller is given, we will estimate the region of asymptotic stability and the domains of sliding mode of the closed-loop system.

2. BACKGROUND MATERIAL AND BASIC DEFINITIONS

An important concept in VSC is that of an attractive manifold on which certain desired dynamical behaviors are guaranteed. Trajectories of the system should be steered towards the manifold and subsequently constrained to remain on it.

We next describe the manifold which is used in the paper.

Suppose

$$S = \begin{bmatrix} s_1 \\ \vdots \\ s_m \end{bmatrix} \in \mathbb{R}^{m \times n},$$

where

---

* The work of these authors was supported by the School of Electrical Engineering of Purdue University.

** The work of this author was partially supported by the National Science Foundation.

$$s_i \in \mathbb{R}^{1\times n} .$$

We assume that S is of full rank.

Let

$$\sigma(x) = \begin{bmatrix} \sigma_1(x) \\ \vdots \\ \sigma_m(x) \end{bmatrix} = \begin{bmatrix} s_1 x \\ \vdots \\ s_m x \end{bmatrix} = Sx ,$$

where

$$x \in \mathbb{R}^n ,$$

and let

$$\Omega = \{x \mid \sigma(x) = 0\} . \tag{2.1}$$

Consider the system

$$\dot{x} = Ax + Bu , \tag{2.2}$$

where $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $u \in \mathbb{R}^m$. We make the following assumptions:

Assumption 1: The matrix SB is nonsingular.

Assumption 2: The pair (A,B) is completely controllable.

Definition 2.1. The solution of the algebraic equation in u of

$$S\dot{x} = SAx + SBu = 0$$

is called the equivalent control and denoted by u_{eq}, that is,

$$u_{eq} = -(SB)^{-1} SAx .$$

Definition 2.2. The equivalent system is the system that is obtained when the original control u is replaced by the equivalent control u_{eq}, that is,

$$\dot{x} = [I_n - B(SB)^{-1}S]A\,x.$$

3. VSC AND BOUNDED CONTROL ACTION

We assume that the control u in the system (2.2) is bounded and that

$$|u_i| \leqq \mu_i , \quad i = 1,\dots,m , \tag{3.1}$$

where $\mu_i > 0$.

Our goal is to design a controller which satisfies the bounds (3.1) and which induces the sliding mode on Ω in the following sense:

Definition 3.1. ([2])

A domain Δ in the manifold $\sigma = 0$ is a sliding mode domain if for each $\epsilon > 0$ there exists a $\delta > 0$ such that any motion starting in the n-dimensional δ-vicinity of Δ may leave the n-dimensional ϵ-vicinity of Δ only through the n-dimensional ϵ-vicinity of the boundaries of Δ.

In general, the controller in VSC varies its structure depending on the position relative to the switching surface and has the form:

$$u_i = \begin{cases} u_i^+(x) & \text{if} \quad \sigma_i(x) > 0 \\ u_i^-(x) & \text{if} \quad \sigma_i(x) < 0 . \end{cases}$$

It is easy to see that if $\sigma^T \dot{\sigma} < 0$, then the trajectory is tending towards the switching surface. Hence if $\sigma^T \dot{\sigma} < 0$ in a neighborhood of a region Δ of the switching surface, then Δ is a sliding domain ([1], [2]).

Let us assume that the switching surface is chosen so that $SB = I_m$; we will see how this can be accomplished in Section 4. With this assumption, $u_{eq} = -SAx$. We next give a sufficient condition for $\sigma^T \dot{\sigma} < 0$ to hold. Note that

$$\sigma^T \dot{\sigma} = \sigma^T (SAx + SBu) = \sigma^T (-u_{eq} + u)$$

$$= \sum_{i=1}^{m} \sigma_i (-(u_{eq})_i + u_i) .$$

Hence if

$$\begin{cases} u_i^+(x) < (u_{eq})_i & \text{for } \sigma_i(x) > 0 \\ u_i^-(x) > (u_{eq})_i & \text{for } \sigma_i(x) < 0 , \end{cases} \tag{3.2}$$

then $\sigma^T \dot{\sigma} < 0$. In this paper, we use the control law

$$u = \begin{bmatrix} -\mu_1 \operatorname{sgn} \sigma_1 \\ \vdots \\ -\mu_m \operatorname{sgn} \sigma_m \end{bmatrix} . \tag{3.3}$$

With this control law, we have $\sigma^T \dot{\sigma} < 0$ in the region

$$\Omega^* = \bigcap_{i=1}^{m} \{x \mid |(u_{eq})_i| < \mu_i\} = \bigcap_{i=1}^{m} \{x \mid |s_i Ax| < \mu_i\} .$$

Note that Ω^* is an open neighborhood of the origin. If $\Delta = \Omega \cap \Omega^*$, where $\Omega = \{x \mid \sigma(x) = 0\}$, then Δ is a sliding domain.

4. DESIGN OF THE SWITCHING HYPERPLANE

There are two basic steps in the design of VSC:

(1) The design of the switching surface so that the behavior of the system has certain prescribed properties on the surface. For example, the switching surface will be designed so the system is asymptotically stable on the surface.

(2) The design of the control strategy to steer the system to the switching surface and to maintain it there.

In this Section we will discuss a method for designing of the switching surface. The method is based on that in [5].

Consider the equivalent system

$$\dot{x} = [I_n - B(SB)^{-1}S]Ax . \tag{4.1}$$

It is easy to see that $B(SB)^{-1}S$ is a projector and has rank m. Hence $I_n - B(SB)^{-1}S$ is also a projector with rank $n-m$ ([8]). Therefore the matrix $A_{eq} = [I_n - B(SB)^{-1}S]A$ in the equivalent system can have at most $n-m$ nonzero eigenvalues. Our goal is to choose S so that the nonzero eigenvalues of A_{eq} are prescribed negative real numbers $\{\lambda_1, ..., \lambda_{n-m}\}$ and the corresponding eigenvectors $\{w_1, ..., w_{n-m}\}$ are to be chosen. Let $W = [w_1 ... w_{n-m}]$; note that $W \in \mathbb{R}^{n \times (n-m)}$. In sliding mode, the system is described by

$$\begin{aligned} \dot{x} &= A_{eq} x \\ \sigma(x) &= Sx = 0 . \end{aligned} \tag{4.2}$$

Denote by R(T) the range of the operator T. Since we require SB to be nonsingular and SW = 0, we must have

$$R(B) \cap R(W) = \{0\} .$$

It then follows that ([5]) we should choose the generalized inverses B^g, W^g of B, W so that

$$B^g W = 0 \tag{4.3a}$$

and

$$W^g B = 0 . \tag{4.3b}$$

We choose $\{w_1, \ldots, w_{n-m}\}$ so that (4.3b) holds. We can now construct S. Let $W^{\perp} \in \mathbb{R}^{m \times n}$ be any full rank annihalator of W, that is $W^{\perp} W = 0$. Since a necessary condition for Sx = 0 to be a switching surface is SW = 0, we see that $\Gamma W^{\perp}$ for any nonsingular $\Gamma \in R^{m \times m}$ is a candidate. We also require that $SB = I_m$ (see Section 3). Note that since $R(W) \cap R(B) = \{0\}$, $W^{\perp} B$ is invertible. We let $\Gamma = (W^{\perp} B)^{-1}$ and let $S = \Gamma W^{\perp}$. It is easy to see that $SB = I_m$ and hence $(W^{\perp} B)^{-1} W^{\perp}$ is a generalized inverse of B. If we let $B^g = S$ in (4.3a), the condition is satisfied.

5. DECOUPLING OF THE FAST AND SLOW MOTIONS

In this Section we introduce a transformation which decouples the system into the fast and slow subsystems, that is, the systems which govern the dynamics of the system off the switching surface and on the switching surface. See [9] for a different approach.

Let $M \in \mathbb{R}^{n \times n}$ be defined by

$$M = \begin{bmatrix} W^g \\ S \end{bmatrix} .$$

Note that M is invertible with $M^{-1} = [W \;\; B]$. Introduce the new coordinates

$$\hat{x} = Mx .$$

Let $z = W^g x$ and $y = Sx$. Then $\hat{x} = \begin{bmatrix} z \\ y \end{bmatrix}$. In the new coordinates, the system becomes

$$\dot{\hat{x}} = MAM^{-1}\hat{x} + MBu .$$

We write

$$MAM^{-1} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

where $A_{11} \in \mathbb{R}^{(n-m)\times(n-m)}$, $A_{22} \in \mathbb{R}^{m \times m}$. Note that

$$MB = \begin{bmatrix} 0 \\ I_m \end{bmatrix} .$$

Hence

$$\dot{z} = A_{11} z + A_{12} y$$
$$\dot{y} = A_{21} z + A_{22} y + u .$$

Observe that $y = \sigma$ and that $A_{21} z + A_{22} y = SAM^{-1}\hat{x} = SAx = -u_{eq}$. The slow subsystem is governed by the first equation and the fast subsystem is governed by the second equation which we can write as:

$$\begin{cases} \dot{z} = A_{11} z + A_{12} \sigma \\ \dot{\sigma} = - u_{eq} + u . \end{cases} \tag{5.1}$$

Observe that $(A - BK)W = WJ$, where $J = \mathrm{diag}\,[\lambda_1,\ldots,\lambda_{n-m}]$, and $K = (SB)^{-1}SA$. Hence

$$W^g A W = J$$

since $W^g B = 0$ and $W^g W = I$. We know that $A_{11} = W^g A W$, and therefore $A_{11} = J$.

6. ESTIMATION OF STABILITY AND SLIDING DOMAINS

In this Section we work in the new coordinates introduced in Section 5 and the system is given by (5.1). We analyze the closed-looped system (5.1) with the controller (3.3). For convenience, we let

$$D = \mathrm{diag}[\mu_1,\ldots,\mu_m] \tag{6.1}$$

and

$$\mathrm{sgn}\sigma = \begin{bmatrix} \mathrm{sgn}\sigma_1 \\ \vdots \\ \mathrm{sgn}\sigma_m \end{bmatrix}.$$

We can now write (3.3) as

$$u = -D\,\mathrm{sgn}\sigma\,. \tag{6.2}$$

For $i = 1,2$, $j = 1,2$, we let

$$a_{ij} = ||A_{ij}||\,,$$

where the A_{ij}'s are from (5.1) and $||A_{ij}|| = \sup\{||A_{ij}x||_2 \mid ||x||_2 \leqq 1\}$.
Let

$$\mu = \min\{\mu_1,\ldots,\mu_m\}\,,$$

$$\lambda = \min\{\,|\lambda_1|,\ldots,|\lambda_{n-m}|\}\,.$$

Remark: The controllability of the pair (A,B) implies that A_{12} is not zero and hence $a_{12} \neq 0$.

We will consider the cases A_{21} is 0 and A_{21} not necessarily 0 separately in Sections 6.1 and 6.2. The case of $A_{21} = 0$ is simpler and gives a flavor of the argument used in the general case. We obtain explicit bounds on the time it takes to reach the switching surface for both cases.

6.1. $A_{21} = 0$

We need the following lemma.

Lemma 6.1. Suppose $\phi(t)$ is real-valued and $k \neq 0$. If

$$\dot{\phi} - k\,\phi \leqq -\mu\,,$$

then for $t \geqq t_0$,

$$\phi(t) \leqq \frac{\mu}{k} + (\phi(t_0) - \frac{\mu}{k})\,e^{k(t-t_0)}\,.$$

Proof. Note that $\dot{\phi} - k\,\phi \leqq -\mu$ is equivalent to $\frac{d}{dt}(e^{-kt}\phi) \leqq -\mu e^{-kt}$. The conclusion is obtained by integration.

□

The system has the form:

$$\begin{aligned} \dot{z} &= A_{11}z + A_{12}\sigma \\ \dot{\sigma} &= A_{22}\sigma - D\mathrm{sgn}\sigma\,. \end{aligned} \tag{6.3}$$

Suppose $\sigma(t_0) = \sigma_0 \in \mathbb{R}^m$. If $\sigma \neq 0$, then

$$\frac{d||\sigma||}{dt} = \sigma^T A_{22} \frac{\sigma}{||\sigma||} - \frac{1}{||\sigma||} \sum_{i=1}^{m} \mu_i \ |\sigma_i| \ .$$

Since $\sum_{i=1}^{m} |\sigma_i| \geqq ||\sigma||$, we have

$$\frac{d||\sigma||}{dt} \leqq ||A_{22}|| \ ||\sigma|| - \mu = a_{22}||\sigma|| - \mu \ .$$

Note that if $0 < ||\sigma|| < \frac{\mu}{a_{22}}$, then $\frac{d||\sigma||}{dt} < 0$. Hence $\sigma(t_1) = 0$ implies that $\sigma(t) = 0$ for $t \geqq t_1$. Also $\frac{d||\sigma||}{dt} < 0$ is equivalent to the condition $\sigma^T \dot{\sigma} < 0$ we have in Section 3. By Lemma 6.1,

$$||\sigma(t)|| \leqq \frac{\mu}{a_{22}} + (||\sigma_0|| - \frac{\mu}{a_{22}}) \ e^{a_{22}(t-t_0)} \ .$$

Therefore if $||\sigma_0|| < \frac{\mu}{a_{22}}$, then $\sigma(t)$ is 0 for some finite t with

$$t \leqq t_0 + \frac{1}{a_{22}} \left[\log\mu - \log(\mu - a_{22}||\sigma_0||) \right] . \tag{6.4}$$

Remark: If A_{22} is stable, then $\mathbb{R}^n$ is a region of asymptotic stability and the switching surface is reached in finite time. Otherwise, a region of asymptotic stability is given by $\{x \in \mathbb{R}^n : ||Sx|| < \frac{\mu}{a_{22}}\}$. In both cases, $\sigma = 0$ is a sliding domain.

6.2. The general case when A_{21} is not necessarily zero

We have

$$\dot{z} = A_{11} z + A_{12} \sigma$$
$$\dot{\sigma} = A_{21} z + A_{22} \sigma - D \text{sgn} \sigma \ .$$

Suppose $||z|| \neq 0$, $||\sigma|| \neq 0$. Then

$$\frac{d||\sigma||}{dt} = \frac{\sigma^T \dot{\sigma}}{||\sigma||} = \frac{\sigma^T}{||\sigma||} A_{21} z + \sigma^T A_{22} \frac{\sigma}{||\sigma||} - \sum_{i=1}^{m} \mu_i \frac{|\sigma_i|}{||\sigma||} \ ,$$

and

$$\frac{d||z||}{dt} = \frac{z^T \dot{z}}{||z||} = z^T A_{11} \frac{z}{||z||} + \frac{z^T}{||z||} A_{12} \sigma \ .$$

We obtain

$$\frac{d||\sigma||}{dt} \leqq a_{21} \ ||z|| + a_{22} \ ||\sigma|| - \mu \tag{6.5}$$

and

$$\frac{d||z||}{dt} \leqq - \lambda \ ||z|| + a_{12}||\sigma|| \ . \tag{6.6}$$

Let

$$\Sigma_1 = \{(z,\sigma) \mid ||\sigma|| < \frac{\lambda\mu}{a_{12} a_{21} + \lambda a_{22}} \ , \ z \in \mathbb{R}^{n-m} \ , \ \sigma \in \mathbb{R}^m\} \ ,$$

$$\Sigma_2 = \{(z,\sigma) \mid a_{21}||z|| + a_{22}||\sigma|| < \mu \ , \ z \in \mathbb{R}^{n-m} \ , \ \sigma \in \mathbb{R}^m\} \ ,$$

and

$$\Sigma = \Sigma_1 \cap \Sigma_2 \ .$$

Theorem 6.2. A trajectory that starts in Σ stays in Σ and reaches the switching surface in finite time, which implies that Σ is a region of asymptotic stability (RAS).

Proof. Let

$$N_1 = \{(z,\sigma) \mid a_{12}\|\sigma\| \geqq \lambda\|z\|\}$$

and

$$N_2 = \{(z,\sigma) \mid a_{12}\|\sigma\| < \lambda\|z\|\} . \qquad \text{(see Fig. 1.)}$$

By (6.5) we have $\frac{d\|\sigma\|}{dt} < 0$ in Σ. Hence if $(z(t_0), \sigma(t_0)) \in \Sigma$, we have $(z(t), \sigma(t)) \in \Sigma_1$ for $t > t_0$. For $(z(t),\sigma(t)) \in N_1 \cap \Sigma \subset N_1 \cap \Sigma_1$, we have

$$a_{21}\|z\| + a_{22}\|\sigma\| \leqq \frac{a_{21}a_{12}}{\lambda}\|\sigma\| + a_{22}\|\sigma\| < \frac{a_{21}a_{12}}{\lambda}\|\sigma(t_0)\| + a_{22}\|\sigma(t_0)\| .$$

Therefore

$$a_{21}\|z\| + a_{22}\|\sigma\| < \mu - \epsilon \quad \text{for some } \epsilon > 0 .$$

Thus we can conclude that a trajectory $(z(t), \sigma(t))$ can leave Σ only through $N_2 \cap \Sigma$. However we have $\frac{d\|\sigma\|}{dt} < 0$ in Σ and $\frac{d\|z\|}{dt} < 0$ in N_2, and hence $a_{21}\|z\| + a_{22}\|\sigma\|$ is a decreasing function in $N_2 \cap \Sigma$. Therefore a trajectory cannot leave $N_2 \cap \Sigma$. Hence a trajectory which starts in Σ stays in Σ.

Suppose $(z(t), \sigma(t)) \in N_1 \cap \Sigma$ for $t_1 \leqq t \leqq t_2$. Then we have

$$\begin{aligned} \frac{d\|\sigma\|}{dt} &\leqq a_{21}\|z\| + a_{22}\|\sigma\| - \mu \\ &\leqq \left(\frac{a_{12}a_{21}}{\lambda} + a_{22}\right)\|\sigma\| - \mu . \end{aligned}$$

Let $k = \frac{a_{12}a_{21}}{\lambda} + a_{22}$. Then by Lemma 6.1, we have

$$\|\sigma(t_2)\| \leqq \|\sigma(t_1)\| \, e^{k(t_2-t_1)} - \frac{\mu}{k}\left\{e^{k(t_2-t_1)} - 1\right\} .$$

Therefore

$$\begin{aligned} \|\sigma(t_1)\| - \|\sigma(t_2)\| &\geqq \left(\frac{\mu}{k} - \|\sigma(t_1)\|\right)\left(e^{k(t_2-t_1)} - 1\right) \\ &\geqq \left(\frac{\mu}{k} - \|\sigma(t_1)\|\right) k(t_2 - t_1) \\ &= (\mu - k\|\sigma(t_1)\|)\,(t_2 - t_1) . \end{aligned}$$

Suppose $\sigma(t_0) = \sigma_0$.

Since $\|\sigma(t)\|$ is decreasing in Σ, we have

$$\|\sigma(t_1)\| - \|\sigma(t_2)\| \geqq (\mu - k\,\|\sigma_0\|)\,(t_2 - t_1) .$$

We conclude that a trajectory cannot spend an infinite amount of time in $N_1 \cap \Sigma$ with $\|\sigma(t)\| > 0$.

We claim that if $(z(t_1), \sigma(t_1)) \in N_2 \cap \Sigma$, $(z(t_2), \sigma(t_2)) \in N_2 \cap \Sigma$, and $t_1 < t_2$, then $\|z(t_2)\| < \|z(t_1)\|$. The claim is clear if $(z(t), \sigma(t)) \in N_2 \cap \Sigma$ for $t_1 \leqq t \leqq t_2$. Otherwise, suppose $(z(t), \sigma(t)) \in N_2 \cap \Sigma$ for $t_1 < t \leqq T_1$, $T_2 \leqq t < t_2$, and $(z(T_1), \sigma(T_1))$,

$(z(T_2), \sigma(T_2))$ are on the boundary of $N_2 \cap \Sigma$. Therefore

$$\frac{||z(T_1)||}{||\sigma(T_1)||} = \frac{||z(T_2)||}{||\sigma(T_2)||}.$$

Since $||\sigma(t)||$ decreases in Σ, we have $||z(T_2)|| < ||z(T_1)||$ and we can conclude that $||z(t_2)|| < ||z(t_1)||$ as in the case where the whole segment is in $N_2 \cap \Sigma$.

Hence if $(z(t_0), \sigma(t_0)) \in N_2 \cap \Sigma$, we have for $t \geqq t_0$,

$$\frac{d||\sigma||}{dt} \leqq a_{21}\, ||z(t_0)|| + a_{22}||\sigma(t_0)|| - \mu < 0 .$$

Therefore, if $(z(t), \sigma(t)) \in N_2 \cap \Sigma$ for $t_1 \leqq t \leqq t_2$ then

$$||\sigma(t_1)|| - ||\sigma(t_2)|| \geqq (\mu - a_{21}||z(t_0)|| - a_{22}||\sigma(t_0)||)(t_2 - t_1).$$

We can conclude that a trajectory cannot spend an infinite amount of time in $N_2 \cap \Sigma$ with $\sigma(t) \neq 0$.

Thus we must reach the switching surface $\sigma=0$ in finite time if we start in Σ. □

From the above, we have

Corollary 6.3. Let

$$\beta = \min\left\{\mu - \left(\frac{a_{12}a_{21}}{\lambda} + a_{22}\right)||\sigma(t_0)||,\ \mu - a_{21}||z(t_0)|| - a_{22}||\sigma(t_0)||\right\}$$

Starting at $(z(t_0), \sigma(t_0)) \in \Sigma$, we must reach $\sigma=0$ in

$$t \leqq t_0 + \frac{||\sigma(t_0)||}{\beta} .$$

Using a Lyapunov function argument, we can give another region of asymptotic stability.

Theorem 6.4. A region of asymptotic stability of the system (5.1), (3.3) is

$$\mathscr{R} = \{(z, \sigma)\ |\alpha||z|| + \beta||\sigma|| < \mu\}$$

where $\beta = \frac{1}{2}\left[(a_{22} - \lambda) + \sqrt{(a_{22}+\lambda)^2 + 4a_{12}a_{21}}\right]$ and $\alpha = \beta a_{21}/(\beta+\lambda)$.

Proof. Let V be the positive definite function defined by ([11])

$$V(z, \sigma) = \alpha||z|| + \beta||\sigma|| .$$

The Lyapunov derivative is

$$\dot{V}(z, \sigma) = \alpha\frac{z^T \dot{z}}{||z||} + \beta\frac{\sigma^T \dot{\sigma}}{||\sigma||}.$$

From (6.5), (6.6) we get

$$\dot{V}(z, \sigma) \leqq \alpha(-\lambda||z|| + a_{12}||\sigma||) + \beta(a_{21}||z|| + a_{22}||\sigma|| - \mu)$$

$$= (-\lambda\alpha + \beta a_{21})||z|| + (\alpha a_{12} + a_{22}\beta)||\sigma|| - \mu\beta .$$

Using the values of α and β, we have

$$\dot{V}(z, \sigma) \leqq \beta[\alpha||z|| + \beta||\sigma|| - \mu] .$$

The right hand side is less than 0 in $\mathscr{R}$. This finishes the proof if $z(t) \neq 0$ and $\sigma(t) \neq 0$.

Otherwise observe that $\beta \geqq a_{22}$ and $\alpha \leqq a_{21}$. By (6.5) and (6.6) we have on $\mathscr{R} \cap \{z = 0\}$

$$\frac{d||\sigma||}{dt} \leqq a_{22}\, ||\sigma|| - \mu \leqq \beta\, ||\sigma|| - \mu < 0$$

and on $\mathscr{R} \bigcap \{\sigma = 0\}$

$$\frac{d||z||}{dt} \leqq -\lambda \, ||z|| < 0 \, .$$

Hence $V(z,\sigma)$ decreases on the critical surfaces also and we are done. □

As a consequence of Theorems 6.2 and 6.4 we give a new region of asymptotic stability with sliding.

Theorem 6.5. Let

$$\mathscr{R}_1 = \{(z,\sigma) \, | \alpha||z|| + \beta||\sigma|| < \mu, \; ||\sigma|| \geqq \frac{\lambda}{a_{12}}||z||\}$$

$$\bigcup \; \{(z,\sigma) \, | a_{21}||z|| + a_{22}||\sigma|| < \mu, \; ||\sigma|| \leqq \frac{\lambda}{a_{12}}||z||\} \, .$$

Then $\mathscr{R}_1$ is a region of asymptotic stability with sliding.

Proof. We use the same notation as in Theorems 6.2 and 6.4. Observe that

$$\mathscr{R}_1 = \Sigma \bigcup (\mathscr{R} \bigcap N_1) \, . \qquad \text{(see Fig. 2.)}$$

For a trajectory that starts in $\mathscr{R} \bigcap N_1$ to reach the switching surface $\{\sigma{=}0\}$, it must pass through Σ, which is a region of asymptotic stability with sliding. □

7. CONCLUSIONS

The problem of the estimation of sliding domains and regions of asymptotic stability with sliding for a class of variable structure systems with bounded controllers has been considered in this paper. A transformation which decouples the "fast" and "slow" dynamics was used to facilitate the analysis and Lyapunov type arguments were employed.

An interesting open problem is the application of the approach developed in this paper to high gain feedback controllers discussed in [3] and [4].

REFERENCES

[1] V.I. Utkin, "Sliding Modes and Their Application in Variable Structure Systems," Moscow, Mir Publishers, 1978.

[2] V. I. Utkin, "Variable structure systems with sliding modes," IEEE Trans. Automat. Contr., Vol. AC-22, No. 2, pp. 212-222, 1977.

[3] K.-K.D. Young, P. V. Kokotović, and V. I. Utkin, "A singular perturbation analysis of high gain feedback systems," IEEE Trans. Automat. Contr., Vol. AC-22, No. 6, pp. 931-938, 1977.

[4] V. I. Utkin, "Application of equivalent control method to the systems with large feedback gain," IEEE Trans. Automat. Contr., Vol. AC-23, No. 3, pp. 484-486, 1978.

[5] O. M. E. El-Ghezawi, A. S. I. Zinober and S. A. Billings, "Analysis and design of variable structure systems using a geometric approach," Int. J. Control, Vol. 38, No. 3, pp. 657-671, 1983.

[6] R. A. DeCarlo, S. H. Żak, and G. P. Matthews, "Variable structure control of nonlinear multivariable systems: A tutorial," Proceedings of the IEEE, Vol. 76, No. 3, pp. 212-232, 1988.

[7] W. M. Wonham, "Linear Multivariable Control; A Geometric Approach," New York, Springer Verlag, 1979.

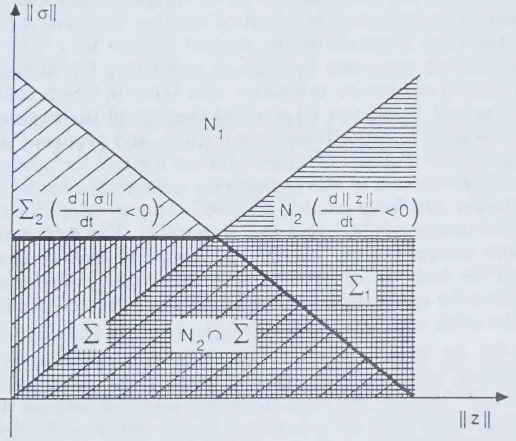
[8] A. Ben-Israel, and T. N. E. Greville, "Generalized Inverses," New York, Wiley, 1974.

[9] P. V. Kokotović, "A Riccati equation for block-diagonalization of ill-conditioned systems," IEEE Trans. Automat. Contr., Vol. AC-20, No. 6, pp. 812-814, 1975.

[10] A. F. Filippov, "Differential equation with discontinuous right-hand sides," American Mathematical Society Translations, Series 2, Vol. 42, pp. 199-231, 1964.

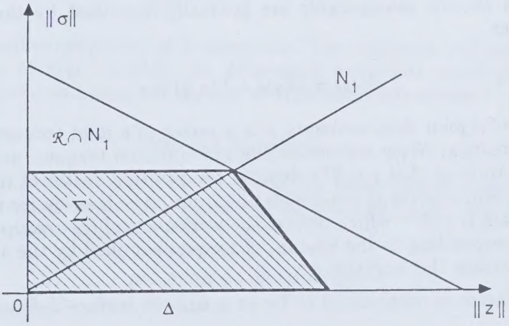
[11] H. H. Rosenbrock, "A method of investigating stability," IFAC Proceedings, pp. 590-594, Basel, Switzerland, 1963.

[12] S. M. Madani-Esfahani and S. H. Żak, "Variable structure control of dynamical systems with bounded controllers," Proc. 1987 American Control Conf., Minneapolis, MN, pp. 90-95, June 10-12, 1987.



$\|\sigma\|$
N_1
$\Sigma_2\left(\frac{d\|\sigma\|}{dt}<0\right)$
$N_2\left(\frac{d\|z\|}{dt}<0\right)$
Σ_1
Σ
$N_2 \cap \Sigma$
$\|z\|$

Fig. 1. Illustration of regions used in the proof of Theorem 6.2.



$\|\sigma\|$
N_1
$\mathcal{R} \cap N_1$
Σ
0
Δ
$\|z\|$

Fig. 2. Illustration of the proof of Theorem 6.5.

The Role of Sliding Mode in Contact Force Control of Robot Manipulators

K. David Young
Lawrence Livermore National Laboratory
University of California
P.O. Box 808, L-97
Livermore, California 94550

Introduction

Many robotic manipulator tasks involve arm motions which are constrained by the task geometry. Such interactions often arise from the necessity or the inevitability to have contacts between the robotic system and its work space. Robot motions which are constrained by the task environment are referred to by Mason [1] as compliant motions. An important design issue in constrained robot dynamics and control involves the control of both the path of the end effector, and the force it exerts on the contact surface by . The so-called hybrid position/force control method, proposed by Raibert and Craig [2], distinguishes the degrees of freedom in which control of position is feasible, and those in which control of force is permissible. The recent work of Yoshikawa [3] clarified the role of the dynamics of the constrained robotic manipulators in hybrid position/force control, and proposed a dynamic hybrid controller whose design is based on the computed-torque method for position control, and on the compensation of the interaction of the end effector constrained force. Khatib [4] approached the constrained motion and force control problem using an operational space formulation in which the computed-torque method was also employed. The dynamics of biped locomotion were formulated and analyzed as constrained mechanical systems by Hemami and Wyman [5], in which the role of the constrained forces on system dynamic stability and control was clearly identified. Techniques for analyzing robotic manipulators subject to constraints were recently proposed by McClamroch and his co-workers [6,7].

In a recent paper [8], we introduced the concept of sliding mode as a tool for analyzing constrained robot motions, and for the design of controllers for steering the end effector from a possibly unconstrained region of the work space to a targeted path on the constrained surface. The work space constraints were not considered in our earlier work [9] which introduced variable structure control and sliding mode to the design of robot control, addressing primarily the design of robot joint servos to track command trajectories. This paper extends the results in [8], and focuses on the role that sliding mode played in the regulation of contact force.

Constrained Robot Dynamics

The dynamics of a robotic manipulator are generally described by the following nonlinear differential equations :

$$J(q)\ddot{q} = G(q)g + F(q,\dot{q}) + u \quad . \tag{1}$$

where q is a vector of n joint displacements, u is a vector of n joint torques, $J(q)$ is the effective moment of inertia matrix, $G(q)g$ represents the gravitational torques, and $F(q,\dot{q})$ denotes the velocity dependent torques. Let $p \in \mathcal{R}^{n_p}$ denotes the position vector of the end effector of the manipulator. If the end effector is not constrained, *i.e.*, it is feasible for p to be any arbitrary vector in a subdomain $\Omega \subset \mathcal{R}^{n_p}$ which defines the workspace of the manipulator, then the joint torque vector u corresponding to the amount of torques supplied by the actuators to move the end effector freely within the workspace.

Suppose the end effector is constrained to be on a smooth surface defined by

$$\phi : \Omega \rightarrow \mathcal{R}^{n_c} , \quad \phi(p) = 0 . \tag{2}$$

The joint torque u then consists of two components:

$$u = u_a + u_r \quad , \tag{3}$$

where u_a is the torque component supplied by the actuators, and u_r is the torque vector corresponding to the reaction force at the end effector due to contact with the constraint surface. The reaction torque u_r can be expressed in terms of the reaction force at the end effector by

$$u_r = K^T(q) f \tag{4}$$

where K is a Jacobian matrix of the function H which defines the transformation between the work space and the joint space :

$$p = H(q) \quad , \tag{5}$$

and

$$K(q) \equiv \frac{\partial H}{\partial q} \quad . \tag{6}$$

Using a Lagrangian formulation of the dynamics of the robotic manipulator subject to a holonomic constraint given by $\phi = 0$, the reaction force f is expressed as

$$f = D(p)^T \lambda \quad , \tag{7}$$

where D is a Jacobian matrix of ϕ

$$D(p) \equiv \frac{\partial \phi}{\partial p} \quad , \tag{8}$$

and λ is the unknown Lagrange multiplier to be determined [10].

Equations (1)-(8) describe the dynamics of a robotic manipulator whose end effector is constrained on the surface $\phi = 0$. Since both unconstrained and constrained motions shall be considered, we need to correlate the equations for the constrained dynamics with that of the unconstrained dynamics. If $\phi \neq 0$, *i.e.*, the end effector is off the surface, then $\lambda = 0$ and $u_r = 0$. Thus with $u = u_a$, Equation (1) is the dynamic model for unconstrained motion.

We shall adopt the usual smoothness assumptions for functions J, D, F, G, H, K and ϕ, *i.e.*, they are at least twice differentiable. It was shown in [5] that if $J(q)$ is positive definite, and $D(H(q))K(q)$ has rank n_c, then the $n_c \times n_c$ matrix defined by

$$A(q) = DKJ^{-1}K^T D^T \tag{9}$$

is positive definite, and consequently it is invertible. The existence and uniqueness of λ result from the invertibility of $A(q)$. In [6,7], the differential equations which govern q and λ were interpreted and analyzed as a singular system of differential equations.

Model Reduction

We introduce a decomposition that is conducive to the reduction of order of the system from $2n$ in the unconstrained mode to $2(n - n_c)$ in the constrained mode. Let the velocity space $\dot{q}$ be decomposed into two subspaces,

$$\dot{q} = J^{-1}\mathcal{N} z + J^{-1}\mathcal{B} A^{-1} y \tag{10}$$

where the columns of $\mathcal{N}$ span the null space of DKJ^{-1} ,and the columns of $\mathcal{B}$ span the range space of $K^T D^T$. Since the matrices J, D and K are functions of q, $\mathcal{N}$ and $\mathcal{B}$ which are also functions of q can be chosen to be

$$DKJ^{-1}\mathcal{N} = 0 \quad ,\mathcal{B} = K^T D^T \tag{11}$$

The two components of $\dot{q}$ in these subspaces are defined by

$$z = \mathcal{M}J\dot{q} \quad \in \mathcal{R}^{n-n_c} \tag{12}$$

$$y = DK\dot{q} \quad \in \mathcal{R}^{n_c} \tag{13}$$

where $\mathcal{M}$ satisfies

$$\mathcal{M}\mathcal{B} = 0 \quad ,\mathcal{M}\mathcal{N} = I \ . \tag{14}$$

As expected, the matrix $\mathcal{M}$ is also a function of q, and to compact notations, the q-dependent matrix functions shall be hereafter abbreviated without the q arguments. Decomposition of this kind has been introduced in [12,13] to reduce the order of dynamic systems involving sliding mode. Equation (1), (3) ,(4) and (7),when expressed in terms of the new variables z and y, becomes

$$\dot{z} = \mathcal{P}(\mathcal{N}z + \mathcal{B}A^{-1}y) + +\mathcal{M}(u_a + Gg + F) \tag{14}$$

$$\dot{y} = \mathcal{Q}J^{-1}(\mathcal{N}z + \mathcal{B}A^{-1}y) + +DKJ^{-1}(u_a + Gg + F) + A\lambda \, , \tag{15}$$

where

$$\mathcal{P} \equiv \dot{\mathcal{M}} + \mathcal{M}\dot{J}J^{-1} \tag{16}$$

$$\mathcal{Q} \equiv \dot{D}K + D\dot{K} \, . \tag{17}$$

Recognizing that

$$\phi = y \quad , \tag{18}$$

and introducing a new "displacement" variable $\psi \in \mathcal{R}^{n-n_c}$ which is defined by

$$\dot{\psi} = \mathcal{M}J\dot{q} = z \quad , \tag{19}$$

complete the construction of a new state space model of the robotic manipulator for both the unconstrained and constrained mode.

For unconstrained robot motion, the reaction force is zero and in view of Equation (7), $\lambda = 0$. Thus, Equations (14) and (15) without the λ term, together with Equations (16) and (17) constitute an alternative set of differential equations describing the unconstrained robot dynamics. For constrained robot motion, $\phi = 0$ and $y = 0$. Hence Equations (16) and (17) are reduced to

$$\dot{z} = \mathcal{P}\mathcal{N}z + \mathcal{M}(u_a + Gg + F) \quad , \tag{20}$$

and together with Equation (18) constitute an $2(n - n_c)$th order differential equation which prescribes the constrained motion on the surface $\phi = 0$. The utility of this model is This model defines two sets of generalized displacements and velocities: one set $\{\phi \, , y\}$ describes the behavior of a motion whose velocity is normal to constrained surface, and the other $\{\psi \, , z\}$ characterizes the motion tangential to the surface. Aside from providing more physical intuitions, this model provides a natural decomposition of the joint torque control into a component to control the unconstrained motion, and another for the constrained motion.

Variable Structure Control Design

The objective of the control system design for constrained robot motion control is to steer the end effector of the robotic manipulator from the unconstrained region of the work space, to a desired path on the constrained surface, while maintaining a desired contact force. The control design proposed in [8] consists of two component control designs: Let

$$u_a = K^T D^T u_f + \mathcal{N} u_c \quad . \tag{21}$$

The design of u_c whose function is constrained motion control is decoupled from that of u_f. In this paper, we shall focus on the design of the component u_f which now plays a dual role of unconstrained motion control and constrained force control: 1) it steers the end effector from an unconstrained mode to a constrained mode, and 2) it produces an equivalent force between the robot and the constrained surface which effectively reduces the contact force to zero.

The component u_f is a variable structure feedback control which is discontinuous on

$$\sigma_f(q, \dot{q}) = C_{f1}\phi(H(q)) + C_{f2}y + \epsilon = 0 \quad , \tag{22}$$

i.e, the ith component of u_f is discontinous on the ith switching hyperplane σ_{f_i} :

$$u_{f_i} = \begin{cases} u_{f_i}^{+}(q, \dot{q}) \quad , & \sigma_{f_i}(q, \dot{q}) > 0 \\ u_{f_i}^{-}(q, \dot{q}) \quad , & \sigma_{f_i}(q, \dot{q}) < 0 \end{cases} \quad , \tag{23}$$

for $i = 1, \ldots, n_c$. The goal in the control design is to induce a sliding mode on $\sigma_f = 0$. That this sliding mode is desirable can be deduced from the sliding mode equation. Suppose sliding mode exists on $\sigma_f = 0$, the resulting sliding mode equation is

$$\dot{\phi} = -C_{f2}^{-1}[C_{f1}\phi + \epsilon] \quad . \tag{24}$$

The matrices defining the set of hyperplanes, C_{f1} and C_{f2}, can be chosen such that $-C_{f2}^{-1}C_{f1}$ is a Hurwitz matrix. Without loss of generality, we let $C_{f2} = I$. If the elements of the vector ϵ are positive, then from the above equation, it can be shown that this sliding mode moves toward the equilibrium point $\phi = -C_{f1}^{-1}\epsilon$, and there exists a finite $T^\star > 0$ such that

$$\phi(T^\star) = 0 \tag{25}$$

and

$$\dot{\phi}(T^\star) = -\epsilon \, . \tag{26}$$

The vector ϵ is introduced to assure that the constrained surface is reached in finite time, *i.e.*, $T^\star < \infty$, furthermore, its value dictates the magnitude of the impact velocity (Equation (26)). An upperbound on the time-to-go until contact while in sliding mode can be estimated using

$$\ln\gamma - \ln[\frac{1}{1 + \phi_d}] \geq \alpha(T^\star - T^r) \, , \tag{27}$$

where T^r denotes the time instant at which the switching hpyerplane $\sigma_f = 0$ is reached, γ is the condition number of eigenvector matrix of C_{f1},

$$\phi_d = \frac{|\phi(T^r)|}{|\phi(T^\star)|} \, , \tag{28}$$

is a measure of the distance-to-go until contact, and

$$\alpha = \max \lambda(-C_{f1}) \, . \tag{29}$$

The contact force resulting from the sliding motion ($\sigma_f = 0$) in contact with the constraint surface ($\phi = 0$) is determined by examining the equivalent control [11] corresponding to the sliding motion before and when contact occurs. Before contact, the unconstrained mode equation of motion applies and by solving for u_f in the equation $\dot{\sigma}_f = 0$, we have

$$u_f^{eq} = -h_R - A^{-1}[QJ^{-1}\mathcal{N}z + (QJ^{-1}\mathcal{B}A^{-1} + C_{f1})(-C_{f1}\phi - \epsilon)] - A^{-1}\dot{\epsilon}\,, \tag{30}$$

where h_R is the component of the gravitational and velocity dependent torques in the range space of $\mathcal{B}$, *i.e.*

$$Gg + F = \mathcal{N}h_N + K^T D^T h_R\,. \tag{31}$$

On the other hand, the contact and applied force when contact occurs are determined from the equation

$$\ddot{\phi} = -\dot{\epsilon}\,, \tag{32}$$

from which we have

$$u_f + \lambda = -h_R - A^{-1}QJ^{-1}\mathcal{N}z + A^{-1}(QJ^{-1}\mathcal{B}A^{-1}\epsilon - \dot{\epsilon})\,. \tag{33}$$

Taking the limit of u_f^{eq} as $\phi \to 0$ in Equation (30) and subsituting into Equation (33) yields

$$\lambda = -A^{-1}C_{f1}\epsilon\,. \tag{34}$$

Thus a zero contact force is acheivable with a choice of ϵ such that its elements are reduced to zero after contact, *e.g.*

$$\epsilon(t) = \begin{cases} \epsilon_0 \quad , & t < T^\star \\ \epsilon_0 e^{\Lambda(t-T^\star)} \quad , Re\lambda(\Lambda) < 0 & t \geq T^\star \end{cases} \tag{35}$$

The physical intepretation of the equivalent control at contact is that it is the same as the reaction force when there is no external torque driving the joints. This conclusion is derived from computing λ from the equation $\ddot{\phi} = 0$ with $u_f = 0$,

$$\lambda = -h_R - A^{-1}QJ^{-1}\mathcal{N}z\,. \tag{36}$$

which is the same expression obtained for u_f^{eq} by taking $\phi \to 0$, $\epsilon \to 0$ and $\dot{\epsilon} \to 0$.

A simple pendulum example

The role of sliding mode in contact force regulation can be illustrated with a simple pendulum whose motion is constrained by a plane perpendicular to the plane of motion. Figure 1 depicts the pendulum and the constraint surface. The dynamic model is given by

$$\dot{x}_1 = \omega \tag{37}$$

$$\dot{\omega} = -\frac{g}{l}\sin(x_1 + \theta_c) + \frac{u_a + \lambda_*}{ml^2}\,, \tag{38}$$

where m is the mass of the attached to the end point of a massless link of length l, and

$$\lambda_* = \begin{cases} 0\,, & x_1 > 0 \\ u_r\,. & x_1 = 0 \end{cases} \tag{39}$$

The constraint surface is a given by

$$x_1 = 0 \quad (\theta = \theta_c)\,, \tag{40}$$

which , using the notations in our constrained robot dynamic model, correpsonds to

$$\phi(p) = p = H(q) \equiv l \sin x_1 = 0\,. \tag{41}$$

The relationship between the reaction torque u_r and reaction force is

$$u_r = l \cos x_1 \,\big|_{x_1=0}\, \lambda = l\lambda\,. \tag{42}$$

Since there is only one degree of freedom ($n = 1$) and one constraint ($n_c = 1$) in this example, no motion is possible once the constraint surface is reached - physically the pendulum stops on a plane which inclines at an angle of θ_c from the stable equilibrium plane. The switching plane for the variable structure control $u_f = u_a$ is chosen as

$$s \equiv \sigma_f = c_1 x_1 + x_2 + \epsilon = 0\,. \tag{43}$$

Figure 2 illustrates the sliding mode on $s = 0$ and the constraint in the phase plane. Computing the equivalent Control directly from $\dot{s} = 0$ yields

$$u_a^{eq} = c_1^2 m l^2 x_1 + mgl \sin(x_1 + \theta_c) + ml^2(C_1\epsilon - \dot{\epsilon})\,. \tag{44}$$

For sliding mode with contact, *i.e.*, $s = 0$ and $x_1 = 0$, we have

$$u_a + u_r = mgl \sin(x_1 + \theta_c) - ml^2\dot{\epsilon}\,. \tag{45}$$

Thus the reaction force due to contact is

$$\lambda = c_1 m l \epsilon\,. \tag{46}$$

If no actuator torque is applied, the reaction force is simply the component of the gravitational force which is tangent to the link, *i.e.*,

$$\lambda = mg \sin \theta_c\,. \tag{47}$$

As ϵ, $\dot{\epsilon} \to 0$,

$$u_a^{eq} \,\big|_{s=0, x_1=0} = mgl \sin \theta_c\,. \tag{48}$$

the equivalent control torque realizes a force to cancel this "natural" contact force, resulting in a zero contact force when contact is made with the inclined plane while in sliding mode.

References

[1] Mason, M. T.,"Compliance and Force Control for Computer Controlled Manipulators", *IEEE Trans. on Systems, Man and Cybernetics*, Vol. SMC-11, pp. 418-432, 1981.

[2] Raibert, M.H. and J.J. Craig, "Hybrid Position/Force Control of Manipulators", *ASME J. Dynamic Systems, Meas. and Control*, Vol.102, pp.126-133, 1981.

[3] Yoshikawa, T., "'Dynamic Hybrid Position/Force Control of Robot Manipulators - Description of Hand Constraints and Calculation of Joint Driving Force," *IEEE Journal of Robotics and Automation*, Vol. RA-3, pp. 386-392, 1987.

[4] Khatib, O., "A Unified Approach for Motion and Force Control of Robot Manipulators: The Operational Space Formulation," *IEEE Journal of Robotics and Automation*, Vol. RA-3, pp. 43-53, 1987.

[5] Hemami, H., and B. F. Wyman,"Modeling and Control of Constrained Dynamic Systems with Applications to Biped Locomotion in the Frontal Plane", *IEEE Trans. Autom. Contr.*, Vol. AC- 24, pp. 526-535, 1979.

[6] McClamroch, N.H. and H.-P. Huang,"Dynamics of a Closed Chain Manipulator", *Proceedings 1985 American Control Conference*, pp. 50-54, 1985.

[7] McClamroch, N. H.,*Singular Systems of Differential Equations as Dynamic Models for Constrained Robot Systems*, Report RSD-TR-2-86, Center for Research on Integrated Manufacturing, The University of Michigan, January 1986.

[8] Young, K. D.,"Application of Sliding Mode to Constrained Robot Motion Control", *Proceedings 1988 American Control Conference, June 15-17, 1988.*

[9] Young, K-K. D., "Controller Design for a Manipulator Using Theory of Variable Structure Systems,", *IEEE Trans. Systems, Man, and Cybernetics*, Vol. SMC-8, 1978, pp. 101-109.

[10] Goldstein, G., *Classical Mechanics*, Second Edition, Addision-Wesley, 1981.

[11] Utkin, V. I.,*Sliding Modes and their Applications*, Moscow : Mir, 1978.

[12] Young, K-K. D., P. V. Kokotovic, and V. I. Utkin,"A Singular Perturbation Analysis of High Gain Feedback Systems", *IEEE Trans. Autom. Contr.*, Vol. AC-22, pp. 931-938, 1977.

[13] V. I. Utkin and Young, K-K. D., "Methods for Construction of Discontinous Planes in Multidimensional Variable Structure Systems", *Automation and Remote Control*, Vol. 39, pp. 1466-1470, 1978.

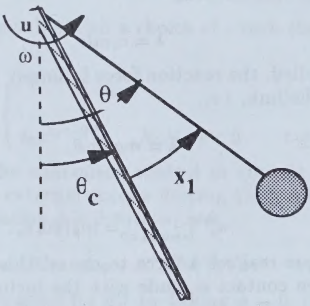


Figure 1: Simple pendulum with constraint

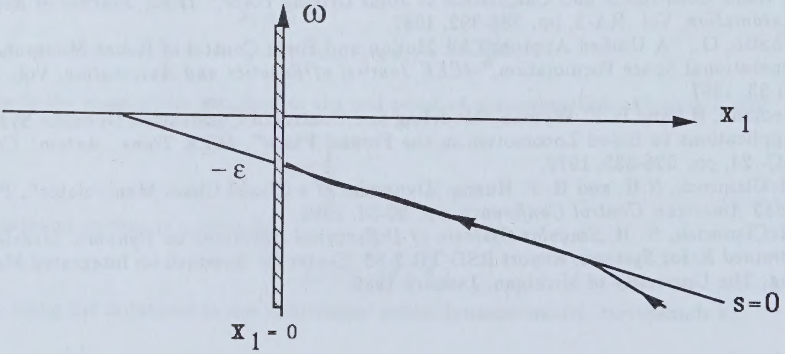


Figure 2: Sliding mode with contact

ON CONTINUOUS APPROXIMATION TO VARIABLE STRUCTURE CONTROL

F. ESFANDIARI and H.K. KHALIL
Department of Electrical Engineering
Michigan State University
East Lansing, MI 48824

Abstract- In this paper we study stability of systems controlled by continuous approximations of variable structure control.

1.INTRODUCTION

Variable Structure Control (VSC) (refer to [1]-[2] for recent surveys) is characterized by control laws that are discontinuous on a certain manifold in the state-space. In practice a discontinuous control is not desirable, since the nonidealities that arise in the implementation of discontinuous control laws (e.g. switching delays) usually result in undesirable features such as chattering, high control activity and excitation of unmodeled high-frequency dynamics. To alleviate these problems, a continuous approximation of the VSC is usually used instead of the VSC itself [3]. In [3] it is shown that if in a boundary layer around the sliding surface the discontinuous control is replaced by a saturation function, then the trajectory of the closed-loop system reaches the boundary layer in a finite time. However this does not ensure stability of the closed-loop system. In this paper we study the stability of the closed-loop system for two different approximations of the VSC: that of [3] and an approximation which is similar to the continuous approximation of min-max control proposed in [4].

2.STABILITY RESULTS

Suppose that the system is in the regular form [1]:

$$\begin{cases} \dot{x}_1 = F_1(x_1, x_2, t) \\ \dot{x}_2 = F_2(x_1, x_2, t) + B(x_1, x_2, t) \left[I_m + \Delta_1(x_1, x_2, t) \right] u + B(x_1, x_2, t)\Delta_2(x_1, x_2, t) \end{cases} \tag{2.1}$$

where x_1, x_2 and u are n-, m- and m- dimensional vectors respectively and $B(x_1, x_2, t)$ is nonsingular for all $x_1 \in \mathbf{R}^n$, $x_2 \in \mathbf{R}^m$ and $t \in \mathbf{R}$. Δ_1 and Δ_2 represent matched uncertainties in $B(x_1, x_2, t)$ and $F_2(x_1, x_2, t)$ respectively.

Assumption 1: $\sqrt{m}\ \| B\Delta_1 B^{-1} \| \le \delta_u < 1$, where δ_u is a nonnegative constant.

Assumption 2: $F_1(\, . \, , \, . \, , \, . \,)$ is globally Lipschitzian in its second argument and there exists a continuously differentiable function $\rho : \mathbf{R}^n \to \mathbf{R}^m$ such that the system $\dot{x}_1 = F_1(x_1, \rho(x_1), t)$ is uniformly exponentially stable and $\rho(0_n) = 0_m$.

Let the VSC be [1]

$$u = -\,\phi(x,t)\, B^{-1}\, sgn(s), \quad s := x_2 - \rho(x_1), \quad [sgn(s)]_i := sgn\ [s]_i \tag{2.2}$$

$$\text{where} \quad \phi(x,t) \ge \frac{1}{1-\delta_u} \left[\alpha + \| F_2 + B\Delta_2 - \frac{\partial \rho}{\partial x_1} F_1 \| \right] \quad \text{for some } \alpha > 0. \tag{2.3}$$

Then it can be easily checked that, under Assumption 1, $\dot{s} \le -\,\alpha\, \|s\|_1$. Therefore the trajectory reaches the sliding surface in a finite time and on the sliding surface uniform exponential stability of the system is implied by Assumption 2. Now let us replace the signum function in (2.2) by a saturation function of the form [3]:

$$u = -\,\phi(x,t)\, B^{-1}\, sat_\varepsilon(s), \quad [sat_\varepsilon(s)]_i := \begin{cases} sgn\ s_i & \text{if } |s_i| \ge \varepsilon \\ \dfrac{s_i}{\varepsilon} & \text{otherwise} \end{cases} \tag{2.4}$$

Proposition 2.1 summarizes the properties of Control (2.4). The following assumption is needed for the Lyapunov stability analysis of Proposition 2.1.

Assumption 3: The following inequalities hold for all $x \in \mathbf{R}^{m+n}$ and $t \in \mathbf{R}$:

$$\| F_2 + B\Delta_2 - \frac{\partial \rho}{\partial x_1} F_1 \| \le \beta_1 \|x\| + \beta_2 \quad \text{and} \quad \| \rho(x_1) \| \le \beta_3 \|x_1\|$$

This work was supported in part by the National Science Foundation under Grant no. ECS-8610714.

where β_1, β_2 and β_3 are nonnegative constants.

Proposition 2.1: Under Assumptions 1-2, the trajectory of the closed-loop system (2.1) and (2.3)-(2.4) reaches the set Ω_1 in a finite time and remains inside thereafter, where

$$\Omega_1 := \left\{ x \in \mathbf{R}^{m+n} \mid \|s\|_\infty \le \varepsilon \right\}$$

Moreover, if Assumption 3 holds as well, then there exists $\bar{\varepsilon}>0$ and function $h : (0, \bar{\varepsilon}) \to \mathbf{R}^+$ such that $\lim_{\varepsilon\to 0} h(\varepsilon) = 0$ and the closed-loop system is globally uniformly ultimately bounded (G.U.U.B.) [4] with respect to Ω_2 for all $\varepsilon \in (0, \bar{\varepsilon})$, where

$$\Omega_2 := \left\{ x \in \mathbf{R}^{m+n} \mid \|x\| \le h(\varepsilon) \right\}$$

Refer to [5] for the proof.

Proposition 2.1 implies that after a finite time Control (2.4) becomes a linear high-gain control. Therefore it is possible to strengthen the stability result of Proposition 2.1 from G.U.U.B. to exponential stability:

Corollary 2.2: Under Assumptions 1-3, suppose that $\beta_2=0$. Then the closed-loop system (2.1) and (2.3)-(2.4) is uniformly exponentially stable for sufficiently small ε.

Now consider the control

$$u := -\bar{\phi}(x,t)\, B^{-1} N_\varepsilon(s), \quad [N_\varepsilon(s)]_i := \begin{cases} sgn\,(s_i) & \text{if } \bar{\phi}(x,t)\,|s_i| \ge \varepsilon \\ \dfrac{\bar{\phi}(x,t)\, s_i}{\varepsilon} & \text{otherwise} \end{cases} \tag{2.5}$$

$$\text{where} \quad \bar{\phi}(x,t) \ge \frac{1}{1-\delta_u} \left\| \mu\, s + F_2 + B\Delta_2 - \frac{\partial\rho}{\partial x_1} F_1 \right\| \quad \text{for some } \mu > 0. \tag{2.6}$$

The following proposition summarizes the stability properties of Control (2.5).

Proposition 2.3: Under Assumptions 1-2, there exist functions $g : \mathbf{R}^+ \to \mathbf{R}^+$ and $h : \mathbf{R}^+ \to \mathbf{R}^+$ such that

$$\lim_{\varepsilon\to 0} g(\varepsilon) = 0, \quad \lim_{\varepsilon\to 0} h(\varepsilon) = 0$$

and the closed-loop system (2.1) and (2.5)-(2.6) is G.U.U.B. with respect to $\Omega_1 \cap \Omega_2$, where

$$\Omega_1 = \left\{ x \in \mathbf{R}^{m+n} \mid \|s\| \le g(\varepsilon) \right\} \text{ and } \Omega_2 = \left\{ x \in \mathbf{R}^{m+n} \mid \|x\| \le h(\varepsilon) \right\}$$

Refer to [5] for the proof.

The boundary layer used in (2.5) is similar to the continuous approximation of min-max control used in [4]. Comparing (2.4) with (2.5), one can see that the *width* of the boundary layer of (2.5) varies with $\bar{\phi}(x,t)$ as well as ε. Note that unlike Proposition 2.1, Assumption 3 is not needed for the stability analysis of Proposition 2.3. Moreover the closed-loop system of Proposition 2.3 is G.U.U.B. with respect to some compact neighborhood of the origin, regardless of how large ε may be.

REFERENCES

[1] V.I. Utkin, "Discontinuous control systems: State of the art in theory and applications," Proc. IFAC, Frankfurt, 1987.

[2] R.A. Decarlo, S.H. Zak and G.P. Matthews, "Variable structure control of nonlinear multivariable systems: A tutorial," Proceedings of IEEE, vol. 76, no.3, pp. 212-232, 1988.

[3] J.J. Slotine and S.S. Sastry, "Tracking control of nonlinear systems using sliding surfaces, with application to robot manipulators," Int. J. Control, vol. 38, pp. 465-492, 1983.

[4] M. Corless and G. Leitmann, "Continuous state feedback guaranteeing uniform ultimate boundedness for uncertain dynamic systems," IEEE trans. Automat. Control, AC-26, pp. 1139-1144, 1981.

[5] F. Esfandiari and H.K. Khalil, "On continuous approximation to variable structure control," submitted for publication.

STABILITY OF A CLASS OF LINEAR MECHANICAL SYSTEMS WITH LINEAR HOLONOMIC CONSTRAINTS*

Danwei Wang

Program in Computer, Information and Control Engineering
The University of Michigan
Ann Arbor, Michigan 48109

N. Harris McClamroch

Department of Aerospace Engineering
Department of Electrical Engineering and Computer Science
The University of Michigan
Ann Arbor, Michigan 48109

ABSTRACT

For certain a priori assumptions, two necessary and sufficient conditions for asymptotic stability are developed for a class of linear closed loop Lagrangian systems with linear holonomic constraints. The proofs of the two results are based on different approaches to characterizing stable solution flows on a subspace. Possible relaxation of the a priori assumptions is indicated.

INTRODUCTION

We study a class of linear closed loop Lagrangian systems with linear constraints. Such systems have recently been used to describe several classes of constrained robot manipulators [1]. The presence of constraints has been shown to affect the stability of such mechanical systems [2]. It is natural to extend known stability results for unconstrained systems [3,4] to constrained systems.

MAIN RESULTS

Consider the linear differential/algebraic equations:

---

* This material is based on upon work supported by the National Science Foundation under Grant No. MSM-8722266. The Government has certain rights in this material.

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{J}^T\boldsymbol{\lambda} \tag{1}$$

$$\mathbf{J}\mathbf{q} = 0 \tag{2}$$

where $\mathbf{q} \in R^n$, $\dot{\mathbf{q}} \in R^n$, $\ddot{\mathbf{q}} \in R^n$ and $\boldsymbol{\lambda} \in R^m$ are the generalized position, velocity, acceleration and multiplier vectors respectively. **M, C** and **K** are $n \times n$ real constant matrices. **J** is $m \times n$ real constant matrix, where $m < n$.

Differential-algebraic equations in the form (1)-(2) arise in applications [1], so it is natural to consider the equations as given in that form. However, under appropriate assumptions, the equations can be viewed as defining a linear vector field on the subspace $\Omega = \left\{ (\mathbf{q},\dot{\mathbf{q}}) \in R^n \times R^n \mid \mathbf{J}\mathbf{q} = 0, \mathbf{J}\dot{\mathbf{q}} = 0 \right\}$, and it is this fact that is the basis for our subsequent development.

Assumptions: **M** and **K** are assumed to be symmetric and positive definite; **C** is assumed to be symmetric and positive semidefinite; **J** is assumed to have full rank m.

This assumption is somewhat restrictive if viewed in the context of Lagrangian systems, but equations (1)-(2) can also be used to characterize a Lagrangian system with a feedback control loop, i.e., a closed loop Lagrangian system as in [2]. In any event, these assumptions are considerably relaxed in a later part of the paper.

The only equilibrium of this system is $(\mathbf{q},\dot{\mathbf{q}},\boldsymbol{\lambda}) = (0,0,0)$. The system (1) and (2) is said to be asymptotically stable if for any $\varepsilon > 0$ there is a $\delta > 0$ such that if $(\mathbf{q}(0),\dot{\mathbf{q}}(0)) \in \Omega$ and $\|\mathbf{q}(0)\| + \|\dot{\mathbf{q}}(0)\| < \delta$, then the unique solution of (1) and (2) satisfies $\|\mathbf{q}(t)\| + \|\dot{\mathbf{q}}(t)\| < \varepsilon$ for all $t > 0$ and $\|\mathbf{q}(t)\| + \|\dot{\mathbf{q}}(t)\| \to 0$ as $t \to \infty$.

It is easy to see that (1) and (2) are asymptotically stable if the matrix **C** is positive definite, i.e. if there is "complete damping" for the system. But this is certainly not a necessary condition for asymptotic stability since only solutions on the subspace Ω are of interest. We develop weaker conditions on the constraint and the system matrices which guarantee asymptotic stability.

We present two necessary and sufficient conditions for the asymptotic stability of (1) and (2). We first construct an $n \times n$ nonsingular matrix $\mathbf{V} = \left[\mathbf{V}_1 \, , \, \mathbf{V}_2 \right]$ according to the singular value decomposition (SVD) $\mathbf{J} = \mathbf{U}\Sigma\mathbf{V}^T$, where **U** and **V** are orthogonal matrices and $\Sigma = \left[\mathbf{D} \, , \, 0 \right]$, with $\mathbf{D} = diag\,[s_1, \ldots, s_m]$, $s_i > 0$ for s=1,...,m. The $n-m$ columns of the $n \times (n-m)$ matrix $\mathbf{V}_2$ form a basis for the kernel of **J**. Details of the SVD can be found in [5].

We now state the first result.

THEOREM 1:

Under the assumptions stated above, the constrained system (1) and (2) is asymptotically stable if and only if the following $(n-m)^2 \times (n-m)$ *matrix*

$$\mathbf{A} = \begin{bmatrix} \mathbf{V}_2^T\mathbf{C}\mathbf{V}_2 \\ \mathbf{V}_2^T\mathbf{C}\mathbf{V}_2\left[(\mathbf{V}_2^T\mathbf{M}\mathbf{V}_2)^{-1}(\mathbf{V}_2^T\mathbf{K}\mathbf{V}_2)\right] \\ \vdots \\ \mathbf{V}_2^T\mathbf{C}\mathbf{V}_2\left[(\mathbf{V}_2^T\mathbf{M}\mathbf{V}_2)^{-1}(\mathbf{V}_2^T\mathbf{K}\mathbf{V}_2)\right]^{n-m-1} \end{bmatrix} \tag{3}$$

has full rank $n-m$.

Proof:

Introduce the equivalence transformation $\mathbf{q} = \mathbf{V}\mathbf{x}$. We partition $\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}$ where $\mathbf{x}_1$ and $\mathbf{x}_2$ are m and $n-m$ vectors respectively. In the new coordinates the constrained system (1) and (2) can be rewritten as

$$\mathbf{V}_1^T\mathbf{M}\mathbf{V}_2\ddot{\mathbf{x}}_2 + \mathbf{V}_1^T\mathbf{C}\mathbf{V}_2\dot{\mathbf{x}}_2 + \mathbf{V}_1^T\mathbf{K}\mathbf{V}_2\mathbf{x}_2 = \mathbf{V}_1^T\mathbf{J}^T\boldsymbol{\lambda} \tag{4}$$

$$\mathbf{V}_2^T\mathbf{M}\mathbf{V}_2\ddot{\mathbf{x}}_2 + \mathbf{V}_2^T\mathbf{C}\mathbf{V}_2\dot{\mathbf{x}}_2 + \mathbf{V}_2^T\mathbf{K}\mathbf{V}_2\mathbf{x}_2 = 0 \tag{5}$$

$$\mathbf{x}_1 = 0 \tag{6}$$

where (2) is equivalent to (6) since $\mathbf{J}\mathbf{V}_1$ is a $m \times m$ nonsingular matrix; the result $\mathbf{V}_2^T\mathbf{J}^T = 0$ has been used to simplify (5).

The $n-m$ independent variables $\mathbf{x}_2$ are governed by the $n-m$ second order linear ordinary differential equations (5). Therefore the constrained system (1) and (2) is asymptotically stable if and only if (5) is asymptotically stable. From the result by Walker and Schmitendorf [3], (5) is asymptotically stable if and only if the matrix $\mathbf{A}$ in (3) has full rank $n-m$. This completes the proof. ■

Remarks:

1) The condition that the matrix $\mathbf{A}$ in (3) is of full rank is equivalent to: the pair $((\mathbf{V}_2^T\mathbf{M}\mathbf{V}_2)^{-1}(\mathbf{V}_2^T\mathbf{K}\mathbf{V}_2), \mathbf{V}_2^T\mathbf{C}\mathbf{V}_2)$ is observable [4].
2) The proof makes clear that system (1) and (2) is equivalent to a differential equation on the subspace Ω which, in appropriate coordinates, is described by the reduced order differential equations (5).

3) The proof also makes clear that system (1) and (2) is asymptotically stable if and only if the coefficient matrices $\mathbf{V}_2^T\mathbf{M}\mathbf{V}_2$, $\mathbf{V}_2^T\mathbf{C}\mathbf{V}_2$ and $\mathbf{V}_2^T\mathbf{K}\mathbf{V}_2$ guarantee reduced order system (5) is asymptotically stable. Thus the assumptions on symmetryness and positive (semi)definiteness are unnecessary.

4) Each solution of the differential/algebraic equation (1) and (2) is of the form

$$\mathbf{q}(t) = \mathbf{V}_2\mathbf{x}_2(t)$$

$$\boldsymbol{\lambda}(t) = (\mathbf{V}_1^T\mathbf{J}^T)^{-1}\left[\mathbf{V}_1^T\mathbf{M}\mathbf{V}_2\ddot{\mathbf{x}}_2(t) + \mathbf{V}_1^T\mathbf{C}\mathbf{V}_2\dot{\mathbf{x}}_2(t) + \mathbf{V}_1^T\mathbf{K}\mathbf{V}_2\mathbf{x}_2(t)\right].$$

where $\mathbf{x}_2(t)$ is a solution of the ordinary differential equation (5).

We state the second result in terms of the coefficient matrices as follows:

THEOREM 2:

Under the same assumptions stated at the beginning, the system (1) and (2) is asymptotically stable if and only if the following $(m+n(n-m))\times n$ *matrix*

$$\mathbf{B} = \begin{bmatrix} \mathbf{J} \\ \mathbf{C} \\ \mathbf{C}\left\{\left[\mathbf{M}^{-1}\mathbf{J}^T(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)^{-1}\mathbf{J} - \mathbf{I}\right]\mathbf{M}^{-1}\mathbf{K}\right\} \\ \vdots \\ \mathbf{C}\left\{\left[\mathbf{M}^{-1}\mathbf{J}^T(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)^{-1}\mathbf{J} - \mathbf{I}\right]\mathbf{M}^{-1}\mathbf{K}\right\}^{n-m-1} \end{bmatrix} \tag{7}$$

has full rank n.

Proof: Sufficiency:

Choose a Lyapunov function

$$\mathbf{H}(\mathbf{q},\dot{\mathbf{q}}) = \frac{1}{2}\dot{\mathbf{q}}^T\mathbf{M}\dot{\mathbf{q}} + \frac{1}{2}\mathbf{q}^T\mathbf{K}\mathbf{q}.$$

Its time derivative along (1) and (2) is

$$\frac{d}{dt}\mathbf{H}(\mathbf{q},\dot{\mathbf{q}}) = -\dot{\mathbf{q}}^T\mathbf{C}\dot{\mathbf{q}} + \dot{\mathbf{q}}^T\mathbf{J}^T\boldsymbol{\lambda} = -\dot{\mathbf{q}}^T\mathbf{C}\dot{\mathbf{q}}$$

where the second term vanishes since $\mathbf{J}\dot{\mathbf{q}} = 0$.

Since $\mathbf{H}(\mathbf{q},\dot{\mathbf{q}})$ is positive definite and $\frac{d}{dt}\mathbf{H}(\mathbf{q},\dot{\mathbf{q}})$ is nonpositive definite, LaSalle's invariance principle implies asymptotic stability of the equilibrium $(\mathbf{q},\dot{\mathbf{q}},\boldsymbol{\lambda}) = (0,0,0)$ if it can be shown that the largest invariant set of (1) and (2) contained in the set $\left\{(\mathbf{q},\dot{\mathbf{q}},\boldsymbol{\lambda}) : \mathbf{C}\dot{\mathbf{q}} = 0\right\}$ is the equilibrium (0,0,0). In this set, $\mathbf{C}\dot{\mathbf{q}} = 0$, the motion is governed by

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{J}^T\boldsymbol{\lambda} \tag{8}$$

$$\mathbf{J}\mathbf{q} = 0\ . \tag{9}$$

The second derivative of the constraint (9) is zero, i.e.

$$\mathbf{J}\ddot{\mathbf{q}} = -\mathbf{J}\mathbf{M}^{-1}\mathbf{K}\mathbf{q} + \mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T\boldsymbol{\lambda} = 0$$

so that

$$\boldsymbol{\lambda} = (\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)^{-1}\mathbf{J}\mathbf{M}^{-1}\mathbf{K}\mathbf{q}\ .$$

Also, $\mathbf{C}\dot{\mathbf{q}} = 0$ implies that

$$\mathbf{C}\ddot{\mathbf{q}} = \mathbf{C}\left[\mathbf{M}^{-1}\mathbf{J}^T(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)^{-1}\mathbf{J} - \mathbf{I}\right]\mathbf{M}^{-1}\mathbf{K}\mathbf{q} = 0$$

Thus

$$\mathbf{C}\left\{\left[\mathbf{M}^{-1}\mathbf{J}^T(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)^{-1}\mathbf{J} - \mathbf{I}\right]\mathbf{M}^{-1}\mathbf{K}\right\}\dot{\mathbf{q}} = 0$$

Continuing this process, we find

$$\mathbf{C}\left\{\left[\mathbf{M}^{-1}\mathbf{J}^T(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)^{-1}\mathbf{J} - \mathbf{I}\right]\mathbf{M}^{-1}\mathbf{K}\right\}^r\dot{\mathbf{q}} = 0, \qquad r = 0,1,\ldots$$

Since the matrix $\mathbf{B}$ in (7) has full rank n, it follows that $\dot{\mathbf{q}} = 0$. Therefore $\ddot{\mathbf{q}} = 0$ and

$$\mathbf{Kq} = \mathbf{J}^T \boldsymbol{\lambda}$$

$$\mathbf{Jq} = 0$$

This implies $\mathbf{q} = 0$ and $\boldsymbol{\lambda} = 0$. Thus (0,0,0) is asymptotically stable.

Necessity:

We will show that if the matrix $\mathbf{B}$ in (7) does not have full rank n, then there exists a periodic solution which does not converge to the equilibrium (0,0,0).

We first develop two preliminary results needed in the proof.

First, if we write the matrix in (7) as $\mathbf{B} = \begin{bmatrix} \mathbf{J} \\ \mathbf{B}_1 \end{bmatrix}$ and postmultiply by the $n \times n$ nonsingular matrix $\mathbf{V} = \begin{bmatrix} \mathbf{V}_1, \mathbf{V}_2 \end{bmatrix}$, we obtain

$$\mathbf{BV} = \begin{bmatrix} \mathbf{JV}_1 & 0 \\ \mathbf{B}_1\mathbf{V}_1 & \mathbf{B}_1\mathbf{V}_2 \end{bmatrix},$$

since $\mathbf{JV}_2 = 0$. Note that since $\mathbf{JV}_1$ is an $m \times m$ nonsingular matrix, $\mathbf{B}$ is full rank n if and only if the $n(n-m) \times (n-m)$ matrix

$$\mathbf{B}_1\mathbf{V}_2 = \begin{bmatrix} \mathbf{CV}_2 \\ \mathbf{C}\left\{\left[\mathbf{M}^{-1}\mathbf{J}^T(\mathbf{JM}^{-1}\mathbf{J}^T)^{-1}\mathbf{J} - \mathbf{I}\right]\mathbf{M}^{-1}\mathbf{K}\right\}\mathbf{V}_2 \\ \vdots \\ \mathbf{C}\left\{\left[\mathbf{M}^{-1}\mathbf{J}^T(\mathbf{JM}^{-1}\mathbf{J}^T)^{-1}\mathbf{J} - \mathbf{I}\right]\mathbf{M}^{-1}\mathbf{K}\right\}^{n-m-1}\mathbf{V}_2 \end{bmatrix}$$

is full rank. By using the Identity given in the Appendix, the above matrix $\mathbf{B}_1\mathbf{V}_2$ can be rewritten as

$$\mathbf{B}_1\mathbf{V}_2 = \begin{bmatrix} \mathbf{C}\mathbf{V}_2 \\ (-1)\mathbf{C}\mathbf{V}_2\left\{(\mathbf{V}_2^T\mathbf{M}\mathbf{V}_2)^{-1}(\mathbf{V}_2^T\mathbf{K}\mathbf{V}_2)\right\} \\ \vdots \\ (-1)^{n-m-1}\mathbf{C}\mathbf{V}_2\left\{(\mathbf{V}_2^T\mathbf{M}\mathbf{V}_2)^{-1}(\mathbf{V}_2^T\mathbf{K}\mathbf{V}_2)\right\}^{n-m-1} \end{bmatrix} . \qquad \mathbf{(10)}$$

Secondly, we develop a decomposition of R^{n-m}. Since $\mathbf{V}_2^T\mathbf{K}\mathbf{V}_2$ and $\mathbf{V}_2^T\mathbf{M}\mathbf{V}_2$ are symmetric and positive definite, it follows that there are $1 \le l \le n-m$ distinct, real and positive eigenvalues of $(\mathbf{V}_2^T\mathbf{M}\mathbf{V}_2)^{-1}\mathbf{V}_2^T\mathbf{K}\mathbf{V}_2$, denoted by ω_j^2, $j=1,\ldots,l$. Thus, R^{n-m} has the decomposition

$$R^{n-m} = \mathbf{P}_1 \oplus \cdots \oplus \mathbf{P}_l$$

where $\mathbf{P}_j$ $j=1,\ldots,l$ are the invariant subspaces defined by

$$\mathbf{P}_j = \left\{ \mathbf{x} \in R^{n-m} \;:\; (\mathbf{V}_2^T\mathbf{K}\mathbf{V}_2 - \omega_j^2\mathbf{V}_2^T\mathbf{M}\mathbf{V}_2)\mathbf{x} = 0 \right\} .$$

We now complete the proof of necessity. Suppose $\mathbf{B}$ has rank less than n. Then there is $\hat{\mathbf{x}} \in R^{n-m}$, $\hat{\mathbf{x}} \neq 0$ in the null space of $\mathbf{B}_1\mathbf{V}_2$, i.e.

$$\mathbf{C}\mathbf{V}_2\left\{(\mathbf{V}_2^T\mathbf{M}\mathbf{V}_2)^{-1}(\mathbf{V}_2^T\mathbf{K}\mathbf{V}_2)\right\}^p \hat{\mathbf{x}} = 0 \quad , \quad p = 0,1,\ldots,n-m-1 . \qquad (11)$$

In addition,

$$\hat{\mathbf{x}} = \sum_{j=1}^{l} \mathbf{x}_j \quad , \quad \textit{where} \;\; \mathbf{x}_j \in \mathbf{P}_j \;, j = 1,\ldots,l \; .$$

Define the $(n-m)\times l$ matrix $\mathbf{X}$ and the $l\times(n-m)$ matrix $\mathbf{L}$ by

$$\mathbf{X} \triangleq \begin{bmatrix} \mathbf{x}_1, \ldots, \mathbf{x}_l \end{bmatrix} \quad , \quad \mathbf{L} \triangleq \begin{bmatrix} 1 & \omega_1^2 & \cdots & \omega_1^{2(n-m-1)} \\ & & \vdots & \\ 1 & \omega_l^2 & \cdots & \omega_l^{2(n-m-1)} \end{bmatrix} .$$

Then (11) can be rewritten in matrix form as

$$\mathbf{CV}_2\mathbf{XL} = 0 \ .$$

$\mathbf{L}$ has full rank l since ω_j^2, $j=1,...,l$, are distinct. This implies that

$$\mathbf{CV}_2\mathbf{X} = 0$$

or, equivalently, $\mathbf{CV}_2\mathbf{x}_j = 0$ for $j = 1,...,l$. Since $\hat{\mathbf{x}} \neq 0$, there is a $k \in [1,...,l]$ such that

$$\mathbf{x}_k \neq 0 \ , \quad (\mathbf{V}_2^T\mathbf{KV}_2 - \omega_k^2\mathbf{V}_2^T\mathbf{MV}_2)\mathbf{x}_k = 0 \ , \quad \mathbf{CV}_2\mathbf{x}_k = 0 \ .$$

Finally, it is easy to verify that

$$\mathbf{q}(t) = \mathbf{V}_2\mathbf{x}_k \sin(\omega_k t)$$

$$\boldsymbol{\lambda}(t) = (\mathbf{JV}_1)^{-T}\left[\mathbf{V}_1^T\mathbf{KV}_2 - \omega_k^2\mathbf{V}_1^T\mathbf{MV}_2\right]\mathbf{x}_k \sin(\omega_k t) \ .$$

is a periodic solution of (1) and (2). Thus $(\mathbf{q},\dot{\mathbf{q}},\boldsymbol{\lambda}) = (0,0,0)$ is not asymptotically stable. This completes the proof of the theorem. ■

Remarks:

1) The necessary and sufficient condition in Theorem 2 is stated in terms of the coefficient matrices **M**, **C**, **K**, **J**, avoiding use of the SVD, and, therefore, may be easier to check than the condition of Theorem 1.

2) The Lyapunov direct method has been used here since if systems. It is applicable on the invariant subspace $\left\{ (\mathbf{q} \ , \dot{\mathbf{q}}) : \mathbf{Jq} = 0 \ , \mathbf{J}\dot{\mathbf{q}} = 0 \right\}$ where the motion of (1) and (2) is constrained.

Under the assumptions stated at the beginning of the paper, it can be shown that the matrix $\mathbf{A}$ in (3) is full rank iff the matrix $\mathbf{B}$ in (7) is full rank. Hence these two necessary and sufficient conditions can be shown to be equivalent. The proof of their equivalence is omitted because it does not provide any particular insight.

As a consequence of these necessary and sufficient conditions, one interesting corollary is the following:

COROLLARY:

Under the assumptions stated at the beginning, a sufficient condition for asymptotic stability of system (1) and (2) is that the $(n+m)\times n$ *matrix*

$$\begin{bmatrix} J \\ C \end{bmatrix} \tag{12}$$

has rank n.

This is the case where there is "complete damping" in Ω and therefore the system is asymptotically stable. This sufficient condition is much easier to check than the necessary and sufficient conditions presented in Theorem 1 and Theorem 2, especially for systems of high order.

CONCLUSIONS

Under a certain set of assumptions, necessary and sufficient conditions for asymptotic stability have been presented for a class of linear differential-algebraic equations which arise as constrained robot models. One result is stated in terms of a singular value decomposition of the algebraic constraint equation; the other result is stated directly in terms of the specified system matrices. A relaxation of the a priori assumptions, as that the asymptotic stability remains, is also indicated.

In this paper, stability results have been developed for a class of linear closed loop Lagrangian systems with holonomic constraints. As shown in [6], these results can also be applied to characterize the local stability properties, near an equilibrium of a related class of nonlinear closed loop Lagrangian systems with holonomic constraints.

REFERENCES

[1] McClamroch, N.H., "Singular Systems of Differential Equations as Dynamic Models for Constrained Robot Systems," Proceedings of IEEE Conference on Robotics and Automation, April, 1986, San Francisco, Ca.

[2] McClamroch, N. H. and D. Wang, "Feedback Stabilization and Tracking of Constrained Robots", IEEE Transaction on Automatic Control, AC-33, No.5, May, 1988.

[3] Walker, J.A. and Schmitendorf, W.E., "A Simple Test for Asymptotic Stability in Partially Dissipative Symmetric Systems", J. of Applied Mechanics, Trans. of the ASME, Dec. 1973, pp 1120-1121.

[4] Miller, R.K. and Michel, A.N., "Asymptotic stability of Systems: Results Involving the System Topology", SIAM J.Control and Optimization, Vol.18. No.2, March, 1980, pp 181-190.

[5] Klema, V.C. and Laub, A.J., "The Singular Value Decomposition: Its Composition and Some Applications", IEEE Trans. on AC, Vol AC-25 No.2, April, 1980, pp 164-176.

[6] Wang, D. and McClamroch, N.H., "Linearization and Local Stability of Nonlinear Lagrangian Systems with Nonlinear Constraints," in preparation.

APPENDIX

Identity:

Assume **M** *and* **K** *are symmetric and positive definite* $n \times n$ *matrices;* **C** *is symmetric and positive semifinite* $n \times n$ *matrix;* **J** *is* $m \times n$ *matrix with full rank* m *and* $\mathbf{J} = \mathbf{U}\Sigma\mathbf{V}^T$ *is the SVD with the partition* $\mathbf{V} = \left[\mathbf{V}_1, \mathbf{V}_2\right]$*. Then*

$$\left\{\left[\mathbf{M}^{-1}\mathbf{J}^T(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)^{-1}\mathbf{J} - \mathbf{I}_n\right]\mathbf{M}^{-1}\mathbf{K}\right\}^p \mathbf{V}_2$$

$$= (-1)^p \mathbf{V}_2 \left\{(\mathbf{V}_2{}^T\mathbf{M}\mathbf{V}_2)^{-1}\mathbf{V}_2{}^T\mathbf{K}\mathbf{V}_2\right\}^p$$

holds for $p = 0, 1, 2, \cdots$.

TIP POSITION ADAPTIVE CONTROL OF A FLEXIBLE MANIPULATOR

TZUNG-CHENG YANG*, JACKSON C. S. YANG* and PRABHAKAR KUDVA**

* Robotics Lab., Dept. of Mech. Engg., Univ. of Maryland, College Park, MD
** Advanced Technology & Research, Inc., Laurel, MD

ABSTRACT

The present work is to demonstrate the feasibility of an adaptive scheme for the tip position control of a single link flexible manipulator handling unknown changing loads. The control strategy used here is essentially a self-tuning one and consists of system identification and pole-placement control algorithms. The self-tuning scheme is used to take care of any changes in the manipulator dynamic characteristics. However, when a standard self-tuning control scheme is applied to a task dealing with picking up and releasing unknown payloads, undesirable transients occur when the payload is released. In the present work, a new adaptive control algorithm is proposed to eliminate these transients.

SUMMARY

There has been considerable interest in the problem of control of lightweight flexible manipulators. The present trend is clearly towards the application of adaptive schemes in such control [1-4]. Initial computer simulation [1] and experimental [2] results have already demonstrated the feasibility of an adaptive scheme for the tip position control of a flexible arm under no load conditions. The same scheme is experimentally implemented here under a changing load situation especially when a mass is attached to the tip at one position and released at a different position in the same horizontal plane and in quiescent state. This is a realistic situation for any robot and is, therefore, capable of bringing out the intrinsic worth of any control scheme chosen for the flexible manipulator. Picking up or releasing a payload changes the dynamic characteristics of a robot arm. Unless the payload mass is negligible compared to the mass of the overall system, the controller gains would definitely have to be appropriately adjusted to maintain smooth maneuvering. Thus, adaptive control seems to be the most natural choice for manipulating unknown payloads. Even with adaptive control, however, certain specific difficulties have to be addressed. These relate to the possible occurrence of transients (i) when the reference input is changed and (ii) at the instants of payload pick-up or release. The transients due to change in the reference input are governed by the speed of parameter identification, sampling rate and the richness of input excitation. Here, the problem of transients during payload pick-up and release, is discussed.

When a standard self-tuning control scheme was applied, an oscillating transient behavior was found to persist during payload release, but not during payload pick-up. This is attributed to a relatively larger control torque being continued to be generated after payload release, because the control gains are larger than what they should be. The transient subsides when the parameters converge. Such transients are clearly not expected during payload pick-up.

In order to suppress the payload release transient, it was considered to store the parameter estimates prior to the load pick up and to use them to reset the estimates just before releasing the payload. This would ensure that the control gains would be appropriate when the payload is released. Resetting parameter estimates, however, would still give rise to other undesirable transients. This problem is successfully solved by a modified recursive control algorithm, wherein the control signal does not respond immediately to sudden changes in the parameter estimates.

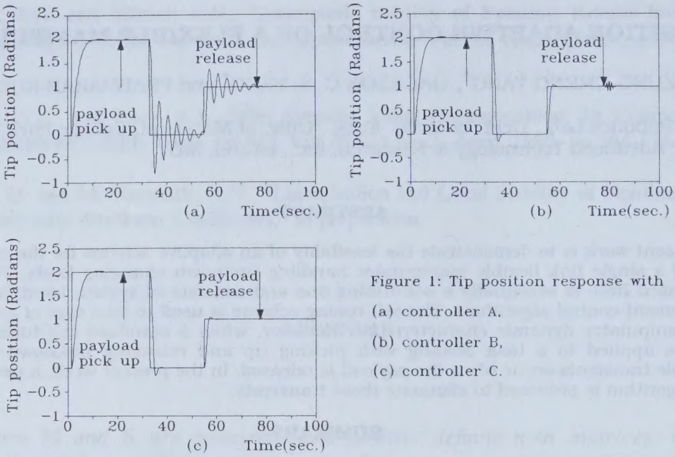


Figure 1: Tip position response with

(a) controller A.

(b) controller B.

(c) controller C.

EXPERIMENTAL RESULTS:

Fig. 1 (a), (b) and (c) are the experimental results for a payload of 52% load factor with the following three controllers:

controller A : constant gain pole-placement controller,
controller B : general adaptive pole-placement controller,
controller C : proposed adaptive pole-placement controller.

The constant gain controller A is seen to perform poorly when the reference input is changed, because the control gains are not appropriate with the payload mass at the manipulator tip. Controller B has a superior performance but for the ***ringing*** effect at payload release. This problem is clearly overcome by controller C. The comparison clearly illustrates the superiority of the proposed adaptive control scheme.

REFERENCES

[1] T. C. Yang, J. C. S. Yang and P. Kudva, "An Adaptive Control Scheme for a Flexible Manipulator," *Proc. 1987 Goddard Conf. on Space Applications of AI and Robotics*, NASA GSFC, Greenbelt, MD, May 1987.

[2] K. H. Sung, T. C. Yang, J. C. S. Yang and P. Kudva, "Experimental Evaluation of an Adaptive Control Scheme for a Flexible Manipulator, " *Proc. 42nd MFPG Meeting*, NBS, Gaithersburg, MD, Sept. 1987.

[3] T. C. Yang, K. H. Sung, J. C. S. Yang and P. Kudva, "An Experimental Adaptive Control Scheme for a Flexible Manipulator," to appear in the *International Journal of Robotics and Automation*, vol. 3, no 2.

[4] D. C. Nemir, A. J. Koivo, R. L. Kashyap, "Pseudolinks and the Self-Tuning Control of a Nonrigid Link Mechanism, " *IEEE Trans. Systems, Man, and Cybernetics*, vol. 18, no 1, pp. 40-48, Jan. 1988.

[5] H. Elliott, and W. A. Wolovich, "Parameter Adaptive Identification and Control," *IEEE Trans. Auto. Control*, vol. AC-24, no. 4, pp. 592-599, Aug. 1979.

[6] H. Elliott, R. Cristi, and M. Das, "Global Stability of Adaptive Pole Placement Algorithms," *IEEE Trans. Auto. Control*, vol. AC-30, no. 4, pp. 348-356, April 1985.

[7] T. R. Fortescue, L. S. Kershenbaum and B. E. Ydstie, "Implementation of Self-tuning Regulators with Variable Forgetting Factors," *Automatica*, vol. 17, no 6, pp. 831-835, 1981.

[8] R. L. Leal and G. C. Goodwin, "A Globally Convergent Adaptive Pole Placement Algorithm without a Persistency of Excitation Requirement," *Proc. 23rd Conference on Decision and Control*, Las Vegas, NV, pp. 669-674, December 1984.

[9] K. H. Sung, P. Kudva and J. C. S. Yang, "Parameter Identification of a Flexible Manipulator," *Proc. IEEE Conference on System, Man and Cybernetics*, Alexandria, VA, October 1987.

COMPENSATORS AND DERIVATIVE ESTIMATION FOR QUEUEING SYSTEMS

MICHAEL A. ZAZANIS
Dept. of Industrial Engineering and Management Sciences
Northwestern University
Evanston, IL 60208

ABSTRACT

A fairly general method for sensitivity analysis of simulations is proposed involving the use of compensator identities. The case of a GI/G/1 queue is discussed in detail and expressions for the sensitivity of state probabilities are derived for the steady state virtual waiting time process and the system size process.

1. Introduction

The problem of sensitivity analysis of simulations has in the last few years attracted the attention of a number of authors. One of the methods proposed, infinitesimal perturbation analysis (e.g. see Ho and Cao, 1983 and Suri and Zazanis, 1988), involves the direct differentiation of sample performance measures. It is well known that, depending on the nature of the parameter and the performance measure, the estimates obtained in this way may be biased (Heidelberger et al., 1988). An alternative method using likelihood ratios was proposed by Reiman and Weiss (1986), Glynn and Sanders (1986), and Rubinstein (1986). The estimators obtained by this method are unbiased for a wide class of systems but in many cases they have large variance. In this paper, a general method for derivative estimation is proposed for a class of problems that can be cast in a form involving stochastic integrals with respect to a point process. It uses compensator identities in conjunction with infinitesimal perturbation analysis (IPA) techniques to provide low variance unbiased estimates at the expense of additional computational requirements.

2. Compensators and sensitivity of simulations

Let $X_t(\theta)$ be a real valued stochastic process depending on a parameter θ, and A_t a point process (e.g. arrivals to or departures from the system), both defined on a filtered probability space and adapted to $\{\mathcal{F}_t\}_{t\geq 0}$. Let λ_t be the $\mathcal{F}_t$ - intensity of A_t and T an a.s. finite stopping time not depending on θ. We will consider performance criteria of the form

$$J(\theta) = E[\int_0^T f(X_s(\theta))\,dA_s] \tag{2.1}$$

and we will derive estimators for the derivative $\frac{d}{d\theta}J(\theta)$. Special emphasis will be given to the case $f(x) = I_B(x)$, where B is a Borel set in R and we will show how to estimate $\frac{d}{d\theta}P_B(\theta)$, where $P_B(\theta)$ is the probability that an arrival finds the system in state B in equilibrium.

The key idea in the method we propose here is the use of the compensator identity

$$E[\int_0^T f(X_s(\theta))\; dA_s\,] = E[\int_0^T f(X_s(\theta))\; \lambda_s ds\,] \tag{2.2}$$

which, assuming X_s to be left continuous, holds for all f for which the expectation exists (e.g. see Bremaud, 1981). While infinitesimal perturbation analysis methods would use $\frac{d}{d\theta}\int_0^T f(X_s(\theta))\; dA_s$ evaluated along a sample path of the process to estimate $\frac{d}{d\theta}J(\theta)$, we use instead $\frac{d}{d\theta}\int_0^T f(X_s(\theta))\; \lambda_s ds$, which in many cases important in applications can easily be evaluated along a sample path. The estimators we propose here have increased computational requirements but are unbiased in many cases where the IPA estimators are biased. The reason for this is that while it may not be permissible to differentiate with respect to θ inside the expectation on the left hand side of (2.2), it is often permissible to do so on the right hand side as we will see in the example given in the following two sections.

A class of algorithms proposed by Gong and Ho (1987) under the name of Smoothed Perturbation Analysis fall for the most part into the framework we present here. However, our approach is more general and rigorous, uses an existing rich mathematical theory, and does not rely on *ad hoc* solutions for each specific problem.

The method we proposed above is illustrated for the case of a GI/G/1 queue.

3. Derivative estimators for steady state probabilities in a GI/G/1 queue

Consider a GI/G/1 queue with input process (S_i, X_i), $i=1, 2, \cdots$, where S_i is the interarrival time between the i^{th} and $(i+1)^{th}$ customers, and $\theta\, X_i$ is the load brought by the i^{th} customer to the system, where θ is a scale parameter varying in some interval $[a,b]$. We will assume that the distribution $G(\ )$ of the interarrival times is absolutely continuous with density $g(\ )$, and that the corresponding hazard rate, $\frac{g(x)}{1-G(x)}$, is bounded by some $M > 0$ for all $x \geq 0$.

Suppose that the first customer arrives to an empty server at time $t = 0$. We will consider the number of customers in the system size process $N_t(\theta)$, as θ varies in an interval $[a,b]$, such that $a(EB_1) < EA_1$ (to ensure stability). Notice that $\{N_t(\theta)\ ;\ t \geq 0\}$ determines $\{N_t(\eta)\ ;\ t \geq 0\}$ for all $\eta \in [a,b]$ (see fig. 1). $\{N_t(\theta)\ ;\ t \geq 0\}$ is defined to be *left continuous.* We will denote by $\{A_s\ ;\ s \geq 0\}$ and $\{D_s\ ;\ s \geq 0\}$ the counting processes associated with the arrival and the departure processes to the system. (These are both defined to be right continuous). Let us define for convenience three additional (left continuous) processes related to $N_s(\theta)$, namely $\{Y_s(\theta)\ ;\ s\geq 0\}$, the age of the busy period process (fig. 2), $\{Z_s\ ;\ s\geq 0\}$, the time since the last arrival process (fig. 3), and $\{\lambda_s\ ;\ s\geq 0\}$, the stochastic intensity process.

Let us denote by $P_k(\theta)$ the probability that an arriving customer in steady state finds k *or more* people in the system (not including himself). Let $T(b)$ be the end of the first busy cycle corresponding to $\theta = b$. Using a standard regenerative argument we can see that

$$P_k(\theta) \;=\; \frac{E[\int_0^{T(b)} I_{\{N_s(\theta)\geq k\}}\, dA_s\,]}{E[\,A_{T(b)} - 1\,]}\;. \tag{3.1}$$

Clearly, $\frac{d}{d\theta}E[A_{T(b)} - 1\,] = 0$ for $\theta < b$ and thus

$$\frac{d}{d\theta}P_k(\theta) \;=\; \frac{1}{E[A_{T(b)} - 1\,]}\;\frac{d}{d\theta}E[\int_0^{T(b)} I_{\{N_s(\theta)\geq k\}}\,\lambda_s ds\,]\;, \tag{3.2}$$

where we have used (2.2). A routine application of the Dominated Convergence Theorem shows that the above can be written as

$$\frac{d}{d\theta}P_k(\theta) \;=\; \frac{1}{E[A_{T(b)} - 1\,]}\;E[\,\frac{d}{d\theta}\int_0^{T(b)} I_{\{N_s(\theta)\geq k\}}\,\lambda_s ds\,] \tag{3.3}$$

and from a sample path argument we get

$$\frac{d}{d\theta}\int_0^{T(b)} I_{\{N_s(\theta)\geq k\}}\,\lambda_s ds \;=\; \frac{1}{\theta}\sum_{i\,\in Q_{k-1}} \lambda_{t_i}\, Y_{t_i} \tag{3.4}$$

where t_i is the time of the i^{th} departure, and Q_{k-1} is the set of indices corresponding to customers that leave the system with $k-1$ customers up to time $T(b)$.

From (3.3), (3.4), and a standard regenerative argument follows that

$$\frac{d}{d\theta}P_k(\theta) \;=\; \frac{1}{\theta}\;\frac{1}{E[\,A_{T(\theta)} - 1\,]}\;E[\sum_{i\in Q_{k-1}} Y_{t_i}\,\frac{g(Z_{t_i})}{1 - G(Z_{t_i})}\,]\;, \tag{3.5}$$

where, in the above expression we have used the fact that λ_{t_i} is simply the hazard rate associated with the renewal arrival process at time t_i. Notice also that $A_{T(\theta)} - 1$ is the number of customers in the first busy period. Finally let us note here that the application of the Dominated Convergence Theorem in (3.2) to get (3.3) would not have been possible had we not used the compensator identity (2.2). In fact, the IPA estimate would be biased in that case.

4. The virtual waiting time process

Consider the same GI/G/1 queue with the slight modification that now θ is the rate with which the server (considered as a deterministic device) performs work. (This simplifies the expressions without affecting the essence of the arguments). Here we will consider the virtual waiting time process V_t^0 (starting with an empty system at t=0), as θ varies in an interval

$[a,b]$, such that $aEB_1 < EA_1$. Thus, the system is stable and a stationary version of the process, $V_t(\theta)$, exists for all $\theta \in [a,b]$ (Loynes, 1962).

Then, if D_x is the number of downcrossings of V_t^0 at level x, t_i the time of the i^{th} downcrossing, and Y_t the age of the busy period process, we can state the following

Theorem: Under the above assumptions and the additional condition that $E[T^2] < \infty$,

$$\frac{d}{d\theta} E[\int_0^T f(V_t^0)\, dA_t] = \int_0^\infty f(dx)\, E[\sum_{i=1}^{D_x} \frac{Y_{t_i}}{\theta} \lambda_{t_i}] \,. \tag{4.1}$$

Proof: We first prove the result for indicator functions of the form $f(y) = I_{(x,\,\infty)}(y)$. We have

$$E[\int_0^T I_{(y,\,\infty)}(V_t^0)\, dA_t] = E[\int_0^T I_{(y,\,\infty)}(V_t^0)\, \lambda_t dt] = E[\int_0^T \sum_{i=1}^{D_x} \int_{s_i}^{t_i} \lambda_t dt] \,, \tag{4.2}$$

where s_i is the time of the i^{th} upcrossing at level x. Differentiating with respect to θ we have

$$\frac{d}{d\theta} E[\int_0^T I_{(y,\,\infty)}(V_t^0(\theta))\lambda_t dt] = \lim_{\delta \to 0} E[\int_0^T \frac{1}{\delta}\{I_{(y,\,\infty)}(V_t^0(\theta+\delta)) - I_{(y,\,\infty)}(V_t^0(\theta))\}\lambda_t dt] \,. \tag{4.3}$$

A straightforward argument shows that $\frac{1}{\delta}|I_{(y,\,\infty)}(V_t^0(\theta+\delta)) - I_{(y,\,\infty)}(V_t^0(\theta))| < t/\theta$ a.s. Moreover, $\lambda_t \leq M$ a.s. for all t by assumption, and an easy application of the Dominated Convergence Theorem gives

$$\frac{d}{d\theta} E[\int_0^T I_{(y,\,\infty)}(V_t^0(\theta))\, dA_t] = E[\frac{d}{d\theta} \sum_{i=1}^{D_y} \int_{s_i}^{t_i} \lambda_t dt] = E[\sum_{i=1}^{D_y} \lambda_{t_i} \frac{dt_i}{d\theta}] \,, \tag{4.4}$$

the last equality being justified by the fact that $\frac{d}{d\theta} s_i = 0$.

In general, let f_n be an increasing sequence of step functions, say $f_n(y) = \sum_{k=1}^{n} a_k I_{(x_k,\,\infty)}(y)$. Then,

$$E[\int_0^T f_n(V_t^0)\, dA_t] = \sum_{k=1}^{n} a_k E[\sum_{i=1}^{D_{x_k}} \int_{s_i(x_k)}^{t_i(x_k)} \lambda_t dt] = \int_0^\infty f_n(dx)\, E[\sum_{i=1}^{D_x} \frac{Y_i}{\theta} \lambda_{t_i}] \,. \tag{4.5}$$

An application of the Monotone Convergence Theorem establishes the theorem in the general case. **QED.**

From the above theorem, we obtain the following useful

Corollary: Let the steady state distribution of the virtual waiting time be $W(x,\theta)$. Let $\overline{W}(x,\theta) = 1 - W(x,\theta)$ denote the associated survivor function. Then,

$$\frac{d}{d\theta}\overline{W}(x,\theta) = -\frac{1}{E[T]}E[\sum_{i=1}^{D_x}\frac{1}{\theta}Y_{t_i}\frac{g(Z_{t_i})}{1-G(Z_{t_i})}], \qquad (4.6)$$

where, as in §3, $g(\ )$ is the density of the interarrival distribution and Z_t the age of the arrival process. (The minus sign here is due to the fact that θ is proportional to the service *rate).*

5. Applications to derivative estimation

In Zazanis and Suri (1986) and Fox and Glynn (1987) it has been shown that finite difference estimates have a very poor performance compared to direct estimates. Here (3.5) and (4.6) suggest such direct estimates for derivatives of the state probabilities of the system size process and the virtual waiting time process in a GI/G/1 queue.

Obviously, while simulating such a system it is very easy to keep track of the quantities that appear in these expressions. In this case of course we take advantage of the fact that the stochastic intensity is simply the hazard rate, and the only part of the history of the process necessary to determine λ_t is the age of the process at time t. For many models used in practice, one would be able to compute the stochastic intensity easily. This would be the case for instance for superpositions of renewal processes (in which case, one would of course need to know all the age processes involved and then would simply add the hazard rates), for Markov renewal processes (in which case one would need to know the state of the underlying chain and the time since the last arrival), for interrupted renewal processes (such as the output from an upstream server in a network without feedback) etc.

REFERENCES

Bremaud, P. (1981). *Point Processes and Queues,* Springer.

Cao, X.R. (1985a). "Convergence of Parameter Sensitivity Estimates in a Stochastic Experiment", IEEE Trans. Aut. Control, Vol.30, No.9, 845-853.

Fox, B.L., and P.W. Glynn (1988). "Replication Schemes for Limiting Expectations", Technical Report #778, SORIE, Cornell.

Glynn, P.W. and J.L. Sanders, (1986). "Monte Carlo optimization of stochastic systems: Two new approaches", *Proceedings of the 1986 ASME Computers in simulation Conference.*

Gong, W.B. and Y.C. Ho (1986). "Smoothed perturbation analysis for discrete event systems", *IEEE Trans. Aut. Control* AC-32,10,pp. 858-866.

Heidelberger, P., X.R. Cao, M.A. Zazanis, and R. Suri (1988). "Convergence Properties of Infinitesimal Perturbation Analysis", to appear in *Management Science*

Ho, Y.C., and X. Cao (1983). "Perturbation Analysis and Optimization of Queueing Networks", *J. Optim. Theory Applic.* *40,* 4, 559-582.

Reiman M.I. and A. Weiss, (1986). "Sensitivity Analysis of Simulations via Likelihood Ratios" Technical Report, AT&T Bell Laboratories.

Rubinstein, R.Y., (1987). "The score function approach for sensitivity analysis of computer simulation models", to appear in *Mathematics and Computers in Simulation.*

Suri, R. and Zazanis, M.A. (1988). "Perturbation Analysis gives strongly consistent sensitivity estimates for the M/G/1 queue", *Management Science 34*, 1, 39-64.

Zazanis, M.A., and R. Suri (1986). "Comparison of Perturbation Analysis with Conventional Sensitivity Estimates for Stochastic Systems", Working Paper # 86-123, Dept. of Ind. Engr., Univ. of Wisconsin-Madison.

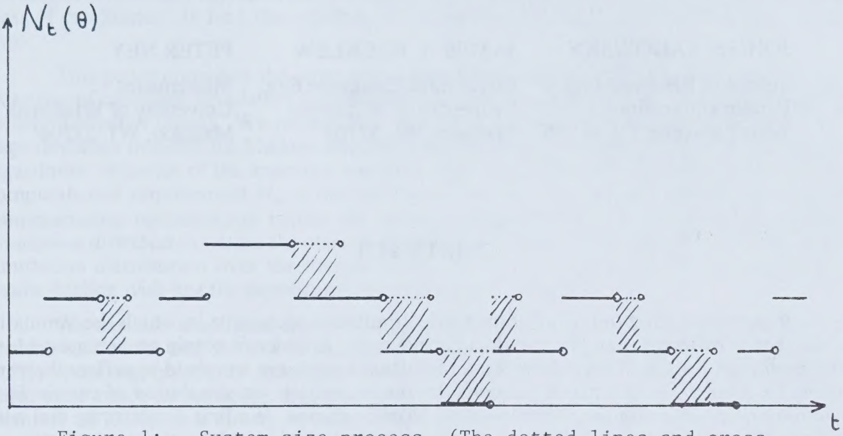


Figure 1: System size process. (The dotted lines and cross-hatched areas added for emphasis correspond to $N_t(\eta)$ for $\eta > \theta$.)

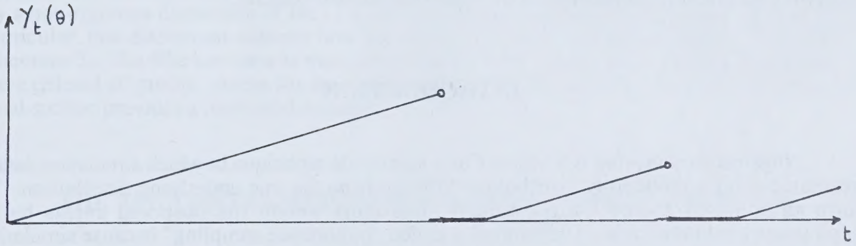


Figure 2: The age of the busy period process.

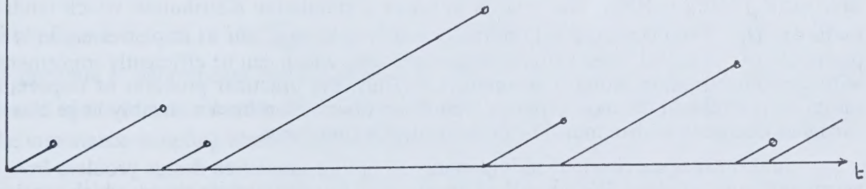


Figure 3: The age of the arrival process.

MONTE CARLO SIMULATION AND LARGE DEVIATIONS THEORY: THE ASYMPTOTIC OPTIMALITY OF THE EXPONENTIAL SHIFT

JOHN S. SADOWSKY
School of Electrical Eng.
Purdue University
West Lafayette, IN 47906

JAMES A. BUCKLEW
Electrical & Computer Eng.
University of Wisconsin
Madison, WI 53706

PETER NEY
Mathematics
University of Wisconsin
Madison, WI 53706

ABSTRACT

Importance sampling is a Monte Carlo simulation technique in which the simulation distribution is different than the true data distribution. In order to obtain an unbiased Monte Carlo estimator for the desired parameter, simulated events are weighted to reflect their true relative frequency. In this paper, we consider the estimation via simulation of certain *large deviation probabilities* for time homogeneous Markov chains. We first demonstrate that when the simulation distribution is also a homogeneous Markov chain, the estimator variance will vanish exponentially as the running sum size n tends to ∞. We then prove that the estimator variance is asymptotically minimized by the same exponentially twisted Markov chain which arises in large deviations theory, and furthermore, *this minimization is unique* among uniformly recurrent homogeneous Markov simulation distributions.

INTRODUCTION

Importance sampling is a Monte Carlo simulation technique in which simulation data is generated using a probability distribution different from the true underlying distribution. To form an unbiased Monte Carlo estimator, one must weight the observed events by an appropriate likelihood ratio. The method is called "importance sampling" because simulation distributions which minimize the estimator variance also tend to increase the relative frequency the "important" events. The efficiency of an importance sampling simulation is usually expressed in terms of computational requirement. Let $\hat{P}_n$ denote the estimator for a probability P_n, and let N_n be the total number of simulations runs required to obtain a specified precision; say, $\mathrm{var}(\hat{P}_n)^{1/2}/P_n = 10\%$. Our goal is to select a simulation distribution which tends to minimize N_n. The unconstrained optimal solution is known, but its implementation is not practical. (A "practical" simulation distribution is one which can be efficiently implemented with computer random number generators.) Thus, the practical problem of importance sampling is to obtain the most efficient simulation distribution from a suitably large class of candidate distributions determined by implementation constraints.

In this paper we consider an importance sampling simulation design problem from an asymptotic point of view. We adopt the framework of *large deviations theory* which considers a sequence of probabilities $\{P_n\}$ vanishing at an exponential rate. As P_n vanishes exponentially fast, it seems quite reasonable that the computational requirement for accurate estimation (that is, N_n) should grow as $n \uparrow \infty$. In fact, N_n *may* grow exponentially fast, and in this case we say the simulation scheme is inefficient. Conversely, we say that *an importance sampling simulation scheme is asymptotically efficient if* N_n *grows less than exponentially* *fast*. In general, there are many efficient simulation schemes, but most of these will not satisfy practical implementation constraints. In various settings, several previous works, including Cottrell, Fort and Malgouyres [CFM], Hunkel and Bucklew [HB], and Siegmund [Si], have considered the family of "exponential shifts" as a *parametric family* of candidate simulation distributions. These works show that the particular exponential shift which occurs in large deviations proofs also asymptotically minimizes estimator variance within the this parametric class of candidates. In fact, this solution is "asymptotically efficient" by our definition of the term.

This paper considers the large deviations setting of time homogeneous Markov sums. Our class of candidate simulation distributions will be the *nonparametric* class of *all* uniformly recurrent Markov chains. We present two new results. Theorem 2 is an importance sampling large deviation theorem for Markov simulation distributions. Theorem 2 precisely identifies the logarithmic behavior of the estimator variance, and hence, also the asymptotic behavior of the computational requirement N_n in the limit as $n \uparrow \infty$. In Theorem 3 we prove the following nonparametric optimization result: *the optimized exponential shift is the unique efficient simulation distribution within the class of Markov candidates*. This uniqueness of the efficient simulation distribution over the nonparametric class is a stronger result than the previous results dealing with just the exponential shifts as a parametric family.

We assume a uniform recurrence hypothesis which is very strong. It allows us to present our results and the key ideas of our proofs with a minimal amount of preliminaries. Ruffly speaking, uniformly recurrent chains are those chains which behave like chains on a finite state space in the sense that they admit a strong Perron-Frobenius theory. Using the methods of [NN], this strong hypothesis can be removed. We do note that the uniform recurrence setting does encompass two important cases: the general i.i.d. case and the finite state space Markov chain case.

The next section provides some background on the general importance sampling problem and the role of Jensen's inequality in the unconstrained optimization. This is followed by a non-rigorous discussion of the i.i.d. case which is intended to highlight our key ideas. In particular, this discussion outlines how the Jensen's inequality is also applied in our proof of Theorem 3. The Markov case is then presented with full rigor. Due to space limitation, we have deleted all proofs, except for the outline provided by the discussion of the i.i.d. case. The final section provides a numerical example.

THE UNCONSTRAINED IMPORTANCE SAMPLING PROBLEM

Let $\{X_k\}$ be a sequence of random variables with distribution $\mathcal{P}(\cdot)$, and define $T_n = 1/n \sum_{k=1}^{n} g(X_k)$. For some Borel set E we wish to estimate the probability

$$P_n = \mathcal{P}(T_n \in E) = \mathcal{P}((X_1,..,X_n) \in E_n) \tag{1}$$

where $E_n = \{ (x_1,..,x_n) : 1/n \sum_{k=1}^{n} g(x_k) \in E \}$. For the purpose of Monte Carlo estimation, we generate N independent realizations of $(X_1,..,X_n)$ using the *simulation distribution* $\mathcal{Q}(\cdot)$ instead of the true distribution $\mathcal{P}(\cdot)$. We assume that $\mathcal{P}|_{E_n} \ll \mathcal{Q}$ where $\mathcal{P}|_{E_n}(A) = \mathcal{P}(A \cap E_n)$. The importance sampling Monte Carlo estimator for P_n is

$$\hat{P}_n = \frac{1}{N} \sum_{i=1}^{N} 1_{E_n}(X_1^{(i)},..,X_n^{(i)}) \frac{d\mathcal{P}|_{E_n}}{d\mathcal{Q}}(X_1^{(i)},..,X_n^{(i)}) \tag{2}$$

where $(X_1^{(i)},..,X_n^{(i)})$ is the i'th independent record of simulation data generated with distribution $\mathcal{Q}(\cdot)$, $1_{E_n}(\cdot)$ is the indicator of the set E_n, and N is the total number of simulation runs. A simple integration indicates that $d\mathcal{P}_{E_n}/d\mathcal{Q}$ is the appropriate weighting factor that makes $\hat{P}_n$ an unbiased estimator for P_n. This is the justification of the absolute continuity requirement that $\mathcal{P}_{E_n} \ll \mathcal{Q}$. Another straightforward integration indicates that the variance of $\hat{P}_n$ is

$$\mathrm{var}[\, \hat{P}_n \,] \;=\; \frac{1}{N} \left\{\, \eta_n(\mathcal{Q}) - P_n^2 \,\right\} \tag{3}$$

where

$$\eta_n(\mathcal{Q}) \;=\; \int_{E_n} \left[\frac{d\mathcal{P}_{E_n}}{d\mathcal{Q}}(x_1,..,x_n) \right]^2 \mathcal{Q}(dx_1,..,dx_n). \tag{4}$$

Notice that the impact of the simulation distribution $\mathcal{Q}(\cdot)$ is entirely contained in the functional $\eta_n(\mathcal{Q})$.

It is our desire to select $\mathcal{Q}(\cdot)$ to minimize $\eta_n(\mathcal{Q})$. The unconstrained optimal solution is known. (See [HH, pp. 57-58]) From Jensen's inequality, we have

$$\eta_n(\mathcal{Q}) \;\geq\; \left\{ \int \left[1_{E_n}(x_1,..,x_n)\, \frac{d\mathcal{P}_{E_n}}{d\mathcal{Q}}(x_1,..,x_n) \right] \mathcal{Q}(dx_1,...,dx_n) \right\}^2 \;=\; P_n^2$$

with $\eta_n(\mathcal{Q}) = P_n^2$ if and only if the $1_{E_n}\, d\mathcal{P}_{E_n}/d\mathcal{Q}$ is constant almost surely with respect to the simulation distribution $\mathcal{Q}(\cdot)$. Integration with respect to $\mathcal{Q}(\cdot)$ determines the normalizing constant to be P_n. The end result is that $\mathrm{var}[\, \hat{P}_n \,] = 0$ if and only if

$$\mathcal{Q}(dx_1,..,dx_n) \;=\; P_n^{-1}\, 1_{E_n}(x_1,..,x_n)\, \mathcal{P}(dx_1,..,dx_n). \tag{5}$$

The unconditionally optimal solution (5) is not practical because it involves precisely the parameter which we do not know, that is, P_n. The variance is zero for just one simulation, however, importance sampling weight is $d\mathcal{P}_{E_n}/d\mathcal{Q} = P_n$ which we do not know. Nonetheless, the result (5) does indicate two interesting properties. First, the "important event" $\{(X_1,..,X_n) \in E_n\}$ occurs with probability 1, and furthermore, the probability of "differential" events like $\{(X_1,..,X_n) \in (dx_1,..,dx_n)\}$ are directly proportional to $\mathcal{P}(dx_1,..,dx_n)$ whenever $(dx_1,..,dx_n) \in E_n$. In practice, it is not possible to exactly duplicate this proportionality on E_n, but a goal should be to approximate it. In the context of this paper, the second property is the really interesting one: *The unconstrained optimal simulation distribution* (5) *does* *not* *preserve independence, stationarity or the Markov property.* In this paper we violate this second property. We shall consider the situation where $\mathcal{P}(\cdot)$ is the distribution of a time homogeneous Markov chain, and we shall restrict attention to candidate simulation distributions which are also homogeneous Markov chains. This is often a practical constraint of computer implementation of $\mathcal{Q}(\cdot)$.

I.I.D. SIMULATION

In order to motivate our general results, we first sketch out the situation for i.i.d. data with i.i.d. simulation distributions. We restrict attention to the case $E = [\gamma,\infty)$ (in this section only), and hence, $P_n = \mathcal{P}(T_n \in E) = \mathcal{P}(T_n \geq \gamma)$. We also suppose that $\gamma > E[g(X_k)]$ so that P_n vanishes and the large deviations limits are non-trivial. Take $\{X_k\text{: } k=0,1,..\}$ to be an i.i.d. sequence with univariate density p(x), and we constraint the simulation distribution to also be i.i.d. with univariate density q(x). We suppose that $p(x) > 0$ whenever $q(x) > 0$ so that $\mathcal{P}|_{E_n} \ll \mathcal{Q}$ for each n, which we recall is required for unbiasedness of the estimator (2). Formula (3) for the estimator variance is still valid, except formula (4) is now replaced by

$$\eta_n(\mathcal{Q}) \;=\; \eta_n(q) \;=\; \int_{E_n} \prod_{k=1}^{n} \left[\frac{p(x_k)}{q(x_k)}\right]^2 q(x_k) \quad dx_1 \ldots dx_n \, . \qquad (6)$$

We begin with an *estimator variance Chernoff bound.* The integrand in (6) is product form. As in the usual derivation of the Chernoff bound for probabilities, we obtain a product form upper bound using an exponential overbound. Specifically, we have $1_{E_n}(x_1,\ldots,x_n) \leq \exp(\alpha[\,\sum_{k=1}^{n} g(x_k) - n\gamma\,])$ for all $\alpha \geq 0$. Thus, we obtain the Chernoff bound

$$\eta_n(q) \;\leq\; \left[\int e^{\alpha[g(x)-\gamma]} \left[\frac{p(x)}{q(x)}\right]^2 q(x)\,dx\right]^n \;=\; \exp(\,-\,n\,[\,\alpha\gamma - \Lambda_q(\alpha)\,]\,) \qquad (7)$$

where $\Lambda_q(\alpha) = \ln(\int e^{\alpha g(x)}\,[p(x)/q(x)]^2\, q(x)\,dx\,)$. Theorem 2 (in the next section) states that when optimized with respect to α, the upper bound (7) is asymptotically tight in the limit as $n \to \infty$, in the logrithmic sense. Specifically, $\eta_n(q)$ *vanishes at an exponential rate* $I_q(\gamma) = \sup_\alpha \{\alpha\gamma - \Lambda_q(\alpha)\}$. This can be proved using standard methods of large deviations theory. (An asymptotic lower bound can is proved by normalizing the "kernel" $e^{\alpha g(x)}\, p(x)^2\, q(x)$ to a probability density and then applying the strong law of large numbers. This corresponds to the "tilted" or "twisted" probability density of the usual i.i.d. large deviations proof.)

The Chernoff bounds and large deviation theory for P_n are well known: P_n vanishes with exponential rate $I(\gamma) = \sup_\alpha \{\alpha\gamma - \Lambda(\alpha)\}$ where $\Lambda(\alpha) = \ln(\int e^{\alpha g(x)}\, p(x)\,dx\,)$. Since variance is non-negative, it follows from (3) that $\eta_n(q) \geq P_n^2$, and hence, in the limit as $n \to \infty$ $\eta_n(q)$ can vanish at an exponential rate no faster than that of P_n^2. Thus, we must have $I_q(\gamma) \leq 2I(\gamma)$. We say that an i.i.d. simulation density q(x) is *efficient* for estimating P_n if $I_q(\gamma) = 2I(\gamma)$.

Our definition of "efficiency" is based on asymptotics in the summation limit n, and not on N_n = the number of simulations used to estimate P_n. However, we can relate this definition to the computational requirement in a very meaningful fashion. As $n \to \infty$, since P_n vanishes exponentially fast it seems reasonable that accurate estimation of P_n should require more and more simulations. To state this precisely, supposed we require the estimator's precision $\mathrm{var}(\hat{P}_n)^{1/2}/P_n$ to be fixed for all n. Let $N_n(q)$ denote the total number of simulation runs required to estimate P_n to the specified precision. Then a simple manipulation indicates that

$N_n(q) \sim e^{Rn}$ for some rate $R \geq 0$. Specifically,

$$R \;=\; \lim_{n \to \infty} \frac{1}{n} \ln(N_n(q)) \;=\; 2\,I(\gamma) - I_q(\gamma) \;\geq\; 0,$$

and $R = 0$ *if and only if* $q(x)$ *is an efficient simulation density.*

Our goal is to asymptotically optimize the simulation density $q(x)$ by minimizing $I_q(\gamma)$, which we shall now do. By replacing γ with a variable t, we deduce that we actually have the inequality $I_q(t) \leq 2I(t)$ for all $t \in \mathbf{R}$. An application of some convex function duality theory indicates that the inequality $I_q(t) \leq 2I(t)$ is equivalent to the inequality $\Lambda_q(2\beta) \geq 2\Lambda(\beta)$ for all $\beta \in \mathbf{R}$, and the equality $I_q(\gamma) = 2I(\gamma)$ holds if and only if $\Lambda_q(2\alpha) = 2\Lambda(\alpha)$ where α is the unique solution of $\Lambda'(\alpha) = \gamma$. This value of α is also the value which optimizes the Chernoff bound for P_n; that is, $I(\gamma) = \alpha\gamma - \Lambda(\alpha)$. Now, from an application of Jensen's inequality we obtain

$$\Lambda_q(2\alpha) \;=\; \ln\Big(\int e^{2\alpha g(x)} \left[\frac{p(x)}{q(x)}\right]^2 q(x)\,dx \Big) \;\geq\; \ln\Big(\left[\int e^{\alpha g(x)} \frac{p(x)}{q(x)}\, q(x)\,dx \right]^2 \Big) \;=\; 2\Lambda(\alpha)$$

and *equality holds if and only if* $e^{\alpha g(x)} p(x)/q(x)$ *is constant.* The constant value is determined by normalization (to make $q(x)$ a probability density). The conclusion is that $q(x) = e^{\alpha g(x) - \Lambda(\alpha)}\, p(x)$ *is the unique efficient i.i.d. simulation density.* This unique solution is precisely the same exponentially "tilted" density which occurs in the proof of the large deviations limit for the sequence of probabilities P_n.

THE MARKOV CASE

Let $\{X_k\text{: } k=0,1,..\}$ be an Markov chain taking values in a measurable state space $(S,\mathcal{S})$ with transition probability $P(x,A)$. Define $T_n = 1/n \sum_{k=1}^{n} g(X_k)$ where $g\text{: } S \to \mathbf{R}$ is a bounded Borel measurable function. We assume hypothesis **H** below, in particular, condition (8) is what we call the *uniform recurrence* condition.

Hypothesis **H**: There exists a probability measure ν_P on $\mathcal{S}$, an integer $m_0 < \infty$ and real numbers $0 < a_P \leq b_P < \infty$ such that

$$a_P\, \nu_P(A) \;\leq\; P^{m_0}(x,A) \;\leq\; b_P\, \nu_P(A) \tag{8}$$

for all $x \in S$, $A \in \mathcal{S}$. Furthermore, letting X be an S-valued random element with distribution ν_P, we assume

$$-\infty \;<\; \underline{g} \;=\; \text{ess inf } g(X) \;<\; \bar{g} \;=\; \text{ess sup } g(X) \;<\; \infty \tag{9}$$

As noted in [INN] and [NN], it is sufficient to work only with the case $m_0 = 1$. Hereafter we implicitly assume $m_0 = 1$.

The following large deviations theorem is an immediate consequence of **H**. It is based on a generalized Perron-Frobenius theorem due originally to T. Harris. See [INN].

Theorem 1: Assume **H**, and for each $\alpha \in \mathbf{R}$ define the non-negative kernel $e^{\alpha g}P(x,dy) = e^{\alpha g(y)}P(x,dy)$. Then $e^{\alpha g}P(x,dy)$ has a unique simple real eigenvalue $\lambda(\alpha)$ and associated eigenfunction $r(x;\alpha)$ with the following properties: for each fixed $\alpha \in \mathbf{R}$, $0 < \lambda(\alpha) < \infty$ and there exists constants c_P, c_P' such that $0 < c_P \leq r(x;\alpha) \leq c_P' < \infty$ for all $x \in S$, and $\Lambda(\alpha) = \ln(\lambda(\alpha))$ is an analytic and strictly convex proper convex function. Define

$$I(t) = \sup_{\alpha \in \mathbf{R}} \{ \alpha t - \Lambda(\alpha) \} \tag{10}$$

for all $t \in \mathbf{R}$. Then I(t) is a non-negative, strictly convex function with I(t) = 0 if and only if $t = \Lambda'(0)$, and for any Borel set E such that

$$\inf_{t \in \bar{E}} I(t) = \inf_{t \in E^o} I(t) = \bar{I}(E) \tag{11}$$

we have

$$\lim_{n \to \infty} \frac{1}{n} \ln(P_n) = -\bar{I}(E). \tag{12}$$

We consider the problem of estimating P_n via importance sampling and we restrict attention to simulation with uniformly recurrent Markov chains generated by a transition probability Q(x,dy). In order to avoid some complexity, we shall also suppose that $X_0 \equiv x_0$ for both the original and simulation distributions. This restriction does not alter the asymptotic results (since $g(\cdot)$ is bounded).

Recall that in order for (2) to yield an unbiased estimator, we required the absolute continuity condition $\mathcal{P}_{E_n} \ll \mathcal{Q}$. Assuming that Q(x,dy) is uniformly recurrent with recurrence measure ν_Q, in the present setting we must have $P(x,\cdot) \ll Q(x,\cdot)$ for all x almost surely with respect to ν_Q. This is necessary in order to have $\mathcal{P}_{E_n} \ll \mathcal{Q}$ for each n. Note that $P(x,\cdot) \ll Q(x,\cdot)$ a.s. ν_Q is equivalent to $\nu_P \ll \nu_Q$, and hence, we hereafter require $\nu_P \ll \nu_Q$.

Define $p_Q(x,y) = \dfrac{dP(x,\cdot)}{dQ(x,\cdot)}(y)$. The variance expression (3) is still valid, except now we replace expression (4) by

$$\eta_n(\mathcal{Q}) = \eta_n(Q) = \int_{E_n} \left[\prod_{k=1}^{n} p_Q(x_{k-1},x_k) \right]^2 Q(x_0,dx_1) \ldots Q(x_{n-1},dx_n). \tag{13}$$

Since $\mathrm{var}[\, \hat{P}_n \,] \geq 0$, from the variance expression (3) it is clear that $\eta_n(Q)$ can vanish no faster than P_n^2. Thus, we have the following immediate consequence of Theorem 1.

Corollary 1: Assume **H** and let Q(x,dy) be any uniformly recurrent transition probability such that $\nu_P \ll \nu_Q$. Then for any Borel set E subject to (11), we have

$$\liminf_{n \to \infty} \frac{1}{n} \ln(\eta_n(Q)) \geq -2\,\bar{I}(E). \tag{14}$$

Definition: We say a uniformly recurrent simulation probability Q(x,dy) is asymptotically *efficient* if $\nu_P \ll \nu_Q$, and

$$\lim_{n\to\infty} \frac{1}{n} \ln(\eta_n(Q)) = -2\,\bar{I}(E). \tag{15}$$

Even though a simulation transition probability satisfies the absolute continuity condition $\nu_P \ll \nu_Q$, and hence, produces an unbiased Monte Carlo estimator, the variance of the resulting estimator may quite bad. In fact, the wrong choice of simulation distribution may lead to an infinite estimator variance. The next lemma identifies another important condition on Q(x,dy) which is required for finite estimator variance.

Lemma 1: Assume **H** and let E be a Borel set which satisfies (11). Also assume that Q(x,dy) satisfies the uniform recurrence condition (8) (with measure ν_Q and constants a_Q and b_Q), and $\nu_P \ll \nu_Q$. If

$$\int \left(\frac{d\nu_P}{d\nu_Q}(y)\right)^2 \nu_Q(dy) < \infty \tag{16}$$

then $\eta_n(Q) < \infty$ for all $n < \infty$. Conversely, if the integral in (16) fails to converge, then $\eta_n(Q) = \infty$ for all n sufficiently large.

We have now identified a suitably large class of candidate simulation transition probabilities Q(x,dy). We need the absolute continuity condition $\nu_P \ll \nu_Q$ in order to have unbiased Monte Carlo estimators. From Lemma 1, we also need condition (16) in order to have finite variance.

Definition 2: $\mathcal{C}$ shall denote the class of all uniformly recurrent Markov transition probabilities Q(x,dy) such that $\nu_P \ll \nu_Q$ and (16) holds.

For all $Q(x,dy) \in \mathcal{C}$, define the non-negative kernel

$$K_Q(x,dy) = p_Q(x,y)^2\, Q(x,dy). \tag{17}$$

With this definition, we rewrite definition (13) as

$$\eta_n(Q) = \int_{E_n} K_Q(x_0,dx_1) \ldots K_Q(x_{n-1},dx_n). \tag{18}$$

In general, $K_Q(x,dy)$ is not a stochastic kernel. However, from (16) it turns out that $K_Q(x,dy)$ does satisfy the uniform recurrence condition (8) with recurrence measure

$$\nu_{K_Q}(dy) = \frac{\left(\frac{d\nu_P}{d\nu_Q}(y)\right)^2 \nu_Q(dy)}{\int \left(\frac{d\nu_P}{d\nu_Q}(z)\right)^2 \nu_Q(dz)}. \tag{19}$$

The associated constants a_{K_Q} and b_{K_Q} can also be expressed explicitly in terms of a_P, b_P, a_Q, b_Q, $\nu_Q(\cdot)$ and $\nu_P(\cdot)$. As a result of the expression (18) and the uniform recurrence of $K_Q(x,dy)$ in (19), we can apply generalized Perron-Frobenius theory to obtain a large deviations theorem for $\eta_n(Q)$. The proof follows standard lines, as in [INN] and elsewhere, and is not effected by the fact that $K_Q(x,dy)$ is not stochastic.

Theorem 2: Assume **H** and consider any $Q \in \mathcal{C}$. For each $\alpha \in \mathbf{R}$ define the non-negative kernel $e^{\alpha g}K_Q(x,dy) = e^{\alpha g(y)}K_Q(x,dy)$. Then $e^{\alpha g}K_Q(x,dy)$ has a unique simple real eigenvalue $\lambda_Q(\alpha)$ and associated eigenfunction $r_Q(x;\alpha)$ with the following properties: for each fixed $\alpha \in \mathbf{R}$, $0 < \lambda_Q(\alpha) < \infty$ and there exists constants c_Q, c_Q' such that $0 < c_Q \leq r(x;\alpha) \leq c_Q' < \infty$ for all $x \in S$, and $\Lambda_Q(\alpha) = \ln(\lambda_Q(\alpha))$ is analytic and strictly convex on **R**. Define

$$I_Q(t) \;=\; \sup_{\alpha \in \mathbf{R}} \{\, \alpha t - \Lambda_Q(\alpha) \,\} \tag{20}$$

for all $t \in \mathbf{R}$. Then $I_Q(t)$ is a strictly convex function which is uniquely minimized at the point $t = \Lambda_Q'(0)$, and for Borel set E subject to condition (11), we have

$$\lim_{n \to \infty} \frac{1}{n} \ln(\, \eta_n(Q)\,) \;=\; -\bar{I}_Q(E) \tag{21}$$

when $\bar{I}_Q(E) = \inf_{t \in E} I_Q(t)$.

We note that $(g,\bar{g}) \subset \{t\colon I(t) < \infty\} = \{t\colon I_Q(t) < \infty\} \subset [g,\bar{g}]$. It follows that (11) also implies to $\inf_{t \in \bar{E}} I_Q(t) = \inf_{t \in E^o} I_Q(t) = \bar{I}_Q(E)$ which is required for the limit (21).

We also note that since $K_Q(x,dy)$ is in general not stochastic, the minimum value $I_Q(\Lambda_Q'(0))$ need not be 0. It will be apparent shortly that $I_Q(\Lambda_Q'(0)) \leq 0$.

We say that a Borel set E has a *dominating point* γ if $\gamma \in \partial E \cap (\underline{g},\bar{g})$ and if either $E \subset [\gamma,\infty)$ and $\Lambda'(0) \leq \gamma$, or $E \subset (\infty,\gamma]$ and $\Lambda'(0) \geq \gamma$ [Ne]. In particular, in the case $\gamma \in (g,\Lambda'(0)]$, γ is the dominating point of $E = (\infty,\gamma]$, and in the case $\gamma \in [\Lambda'(0),\bar{g})$, γ is the dominating point of $E = [\gamma,\infty)$.

Theorem 3: Assume **H** and let $Q(x,dy)$ be any member of the nonparametric class $\mathcal{C}$. Then $I_Q(t) \leq 2I(t)$ for all $t \in \mathbf{R}$, and for any $\gamma \in (g,\bar{g})$ we have $I_Q(\gamma) = 2I(\gamma)$ *if and only if*

$$Q(x,dy) \;=\; e^{\alpha g(y) - \Lambda(\alpha)} \frac{r(y;\alpha)}{r(x;\alpha)} P(x,dy) \tag{22}$$

for all x a. s. ν_P where α is the unique solution of $\Lambda'(\alpha) = \gamma$. In particular, if E has a dominating point γ then *(20) defines the unique efficient simulation transition probability within the admissible class $\mathcal{C}$.*

We note that the proof of Theorem 3 is somewhat more complex than in the i.i.d. case (discussed in the previous section) because in the Markov case we must deal with the eigenfunction as well as the eigenvalue. Nonetheless, the core of the proof of uniqueness is still an application of Jensen's inequality.

A NUMERICAL EXAMPLE

Let $\{X_k: k=0,1,..\}$ be a sequence of i.i.d. random variables with Laplacian probability density function $p_0(x) = 1/2\, e^{-|x+1|}$, and take $g(x) = -1$ for $x \leq -1$, $g(x) = x$ for $-1 < x < 1$, and $g(x) = +1$ for $x \geq 1$. We are interested in estimating the probability $P_n = \mathcal{P}(T_n \geq 0)$, that is, $E = [\gamma,\infty)$ with $\gamma = 0$. This calculation would arise in the computation of the error probabilities for an i.i.d. data Neyman-Pearson test with alternatives $p_0(x)$ vs. $p_1(x) = 1/2\, e^{-|x-1|}$. For i.i.d. data, the eigenvalue $\lambda(\alpha)$ is just the moment generating function of $p_0(x)$, and the asymptotically optimal simulation distribution as identified by Theorem 3 is the i.i.d. distribution with density function $q(x) = e^{\alpha x - \Lambda(\alpha)} p_0(x)$ where α is the unique solution of $\Lambda'(\alpha) = 0$. For the purpose of comparison, we also consider the linearly shifted simulation density $\tilde{q}(x) = p_0(x-c)$. This candidate has recently become quite popular. (See [OA] and references therein, and also see [WHW] for an interesting critique of the linear shift simulation distribution.) From numerical computations of $N_n(\tilde{q})$ we have found that $c = -1$ is most efficient for moderate to large values of n. This is an intuitive choice for c as it puts the mean of T_n at the threshold $\gamma = 0$, which is precisely what the optimized exponential shift does. Hence, $\tilde{q}(x) = 1/2\, e^{-|x|}$. Figure 1 compares the two rate functions, $I_q(t)$ and $I_{\tilde{q}}(t)$, and the rate function upper bound $2I(t)$. The variance rates for estimation of $P_n = \mathcal{P}(T_n \geq 0)$ are $I_q(0) = 2I(0) = 0.6138$ and $I_{\tilde{q}}(0) = 0.5101$. This means that the relative efficiency $N_n(\tilde{q})/N_n(q)$ grows with an exponential rate of 0.1037. On Figure 1, this difference is labeled the "gap" for "performance gap." We performed the computer simulations and empirically determined $N_n(\tilde{q})$ and $N_n(q)$ for a 10% precision. Table 1 presents these computational requirement estimates, along with the estimates of P_n. The estimates of $N_n(\tilde{q})$ and $N_n(q)$ in Table 1 were based enough simulation runs to ensure an accuracy within 10%. The last column of Table 1 is the estimate of $N_n(p) = 100/P_n$ which is the computational requirement for ordinary Monte Carlo with no importance sampling. In the table, it is seen that $N_n(\tilde{q})$ grows exponentially in n, as predicted by our theory. It turns out that $N_n(q)$ behaves like $O(\sqrt{n})$ (see [HB]) and this behavior is also reflected in the empirical data of Table 1. Note that the linear shift $\tilde{q}(x)$ is actually better than the asymptotically optimal exponential shift for $n = 1$. This was also seen in a similar example in [OA]. However, the exponential shift clearly asserts its superiority for $n > 1$.

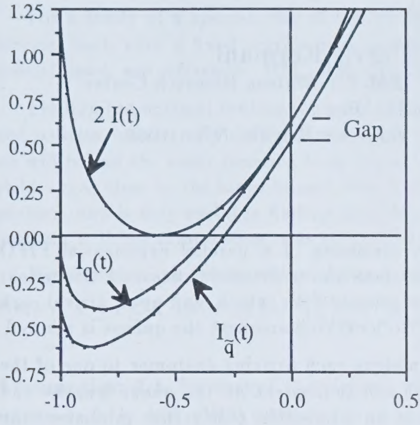


Figure 1: Rate function comparison.

Table 1: Comparison of Computation

n	$\hat{P}_n$	$N_n(q)$	$N_n(\tilde{q})$	$N_n(p)$
1	.189	121	95	529
2	.166	141	246	602
4	.058	203	411	1.7×10^3
6	.027	249	533	3.7×10^3
8	.013	265	751	7.7×10^3
10	.0066	302	1074	1.5×10^4
12	.0031	326	1217	3.2×10^4
14	.0016	348	1629	6.3×10^4
16	.00086	369	2012	1.2×10^5

REFERENCES

[CFM] Cottrell, M., Fort, J-C and Malgouyres, G. (1983) "Large deviations and rare events in the study of stochastic algorithms," *IEEE Trans. Auto. Control,* **AC-28**, pp. 907-920.

[INN] Iscoe, I., Ney, P. and Nummelin, E. (1985) "Large deviations of uniformly recurrent Markov additive processes," *Adv. Appl. Math.*, **6**, pp. 373-411.

[HB] Hunkel, V. and Bucklew, J.A. (1988) "Fast simulation for functionals of Markov chains," *Proc. 22nd Ann. Conf. on Info. Sci. & Sys.*, Princeton University.

[HH] Hammersley, J.M. and Handscomb, D.C. (1964) *Monte Carlo Methods.* New York: Chapman and Hall.

[Ne] Ney, P.(1983) "Dominating points and the asymptotics of large deviations for random walks on R^d," *Ann. Probab.*, **11**, pp. 158-167.

[NN] Ney, P. and Nummelin, E.(1987) "Markov additive process I: eigenvalue properties and limit theorems, II: large deviations," *Ann. Probab.*, **15**, pp. 561-609.

[OA] Orsak, G. and Aazhang, B. (1988) "A comparison of two importance sampling methods for the analysis of detection systems," *Proc. 1988 Conf. Inform. Sci. & Sys.,* Princeton University.

[Si] Siegmund, D. (1976) "Importance sampling in the Monte Carlo study of sequential tests," *Ann. Statis.*, **4**, pp. 673-684.

[WHW] Wessel, A.E., Hall, E.B. and Wise, G.L. (1988) "Some comments on importance sampling," *Proc. 1988 Conf. Inform. Sci. & Sys.*, Princeton University.

CUSTOMER ROUTING TO DIFFERENT SERVERS WITH COMPLETE INFORMATION

Zvi Rosberg
Dept. of Computer Science
Technion - IIT
Haifa 32000, Israel

Parviz Kermani
IBM-T.J. Watson Research Center
P.O. Box 704
Yorktown Heights, N.Y. 10598

Abstract

In this study we consider a queueing system consisting of n parallel exponential FIFO queues, each with its own state dependent service rate and a deterministic state dependent cost rate. Customers arrive according to a Poisson process with rate λ, and upon arrival each customer must be routed to some server's queue. No jockeying amongst the queues is allowed.

An admissible routing policy is a policy that assigns each arriving customer to one of the queues. A decision at time t may be randomized and dependent on the queue lengths and decisions till time t. An optimal routing policy is an admissible policy that minimizes the long-run average holding cost.

In this study, we bound the optimal cost from below, by considering an ideal system, where each server optimally selects a given proportion of customers, irrespective of other servers' selections. From this ideal system we construct a class of admissible routing policies, *the overflow routing class*, that approximates the ideal situation for each server. Finally, we evaluate the policies and compare them to the lower bound.

This talk is based on the research report [1].

Keywords: Exponential Queues, Parallel servers, Customer Routing, Flow control, Markov Decision Processes, Overflow queues.

Consider a queueing system having n exponential servers, each with its own queue and service rate. Customers arrive according to a Poisson process with rate λ, and upon arrival each customer must be routed to some server's queue. No jockeying amongst the queues is allowed and each server serves its queue according to first-come-first-served (FCFS) discipline.

Each exponential server i, $1 \leq i \leq n$, provides service with a state dependent rate $\mu^{(i)}(k)$, $k = 0, 1, \ldots$. That is, $\mu^{(i)}(k)$ is the service rate which is given by server i while there are k customers at queue i. Furthermore, at every queue i, there is a deterministic holding cost which occurs at rate $h^{(i)}(k)$ while there are k customers at the queue.

An admissible policy π is a routing policy that assigns each arriving customer to one of the queues. A routing decision that is taken place at time t, can be randomized and dependent on the queue lengths and decisions until time t. Let $V_T(\pi, \mathbf{k})$ be the total expected holding cost occurred at the system until time T under policy π, given that the initial number of customers at the queues is $\mathbf{k} = (k_1, k_2, \ldots, k_n)$. The long-run average cost of using policy π, given the initial state $\mathbf{k}$, is defined by

$$\bar{V}(\pi, \mathbf{k}) = \limsup_{T \to \infty} \frac{1}{T} V_T(\pi, \mathbf{k}). \tag{1}$$

The optimal long-run average cost (the *value function*) is,

$$\bar{V}(\mathbf{k}) = \inf_{\pi} \bar{V}(\pi, \mathbf{k}).$$

The optimal routing policy π^* (if exists), is the policy that satisfies $\bar{V}(\pi^*, \mathbf{k}) = \bar{V}(\mathbf{k})$, for every state $\mathbf{k}$.

For a study of a special case of this problem, where all queues have identical exponential servers, each with a fixed service rate and holding costs $h^{(i)}(k) = k, \quad 1 \leq i \leq n$, and other special cases, see references [15], [13], [7], [2], [14], [4] [6], [11] and [9].

Finding the optimal routing policy in the general case with different state dependent servers and policies, is extremely difficult (as one can realize from the counterexamples in [14]). Instead, we will bound the *value function* from below, and propose a routing policy that will attempt to yield a cost close to the lower bound. Our bounding technique provides insight into the routing problem and is very useful in finding good heuristic policies. A similar approach for the routing problem without information, was successfully used in [11] and [9].

We assume that the service and the holding cost rates have the following structure. For every i, $1 \leq i \leq n$, let $c^{(i)}(k) = \frac{h^{(i)}(k+1) - h^{(i)}(k)}{\mu^{(i)}(k+1) - \mu^{(i)}(k)}, \qquad k \geq 0$.

Assumption 0.1 *For every i, $1 \leq i \leq n$, $\mu^{(i)}(k)$ and $h^{(i)}(k)$ are non-negative, non-decreasing sequences in k, and $\mu^{(i)}(0) = 0$.*

Assumption 0.2 *For every i, $1 \leq i \leq n$, $c^{(i)}(k)$ is a non-decreasing sequence in k.*

Assumption 0.2 reflects a situation where the holding cost rate increases relatively faster than the service rate.

A special interesting case is the case where $h^{(i)}(k) = k$. Here, the state dependent service rates include a wide class of queueing networks that have a Norton equivalent, [3], [5, pp. 165-170].

To ensure that the value function is finite, we assume the following. First, let $V_T^{(i)}(\pi, \mathbf{k})$ be the total expected cost incurred at server i until time T under policy π, and $\bar{V}^{(i)}(\pi, \mathbf{k}) = \limsup_{T\to\infty} \frac{1}{T} V_T^{(i)}(\pi, \mathbf{k})$. It is well known that under a deterministic splitting routing policy that routes each arriving customer to server i with probability p_i, $\sum_{i=1}^n p_i = 1$,

$$\bar{V}^{(i)}(\pi, \mathbf{k}) = \begin{cases} \sum_{k=0}^{\infty} \xi^{(i)}(k) h^{(i)}(k), & \text{if} \quad \lambda p_i < \lim_{k\to\infty} \mu^{(i)}(k); \\ \lim_{k\to\infty} h^{(i)}(k), & \text{if} \quad \lambda p_i \geq \lim_{k\to\infty} \mu^{(i)}(k), \end{cases} \tag{2}$$

where $\xi^{(i)}(k)$, $k \geq 0$, is the stationary distribution of the birth-death process whose birth rate at state k is λp_i and death rate is $\mu^{(i)}(k)$. That is, $\xi^{(i)}(k) = C^{-1} \prod_{l=0}^{k-1} \frac{\lambda p_i}{\mu^{(i)}(l+1)}$, $k \geq 0$, and C is the finite normalization constant. Now, our third assumption is:

Assumption 0.3 *There exists a vector $\mathbf{p} = (p_1, p_2, \ldots, p_n)$, $p_i \geq 0$, $\sum_{i=1}^n p_i = 1$, and a deterministic splitting routing policy π, under which*

$$\bar{V}^{(i)}(\pi, \mathbf{k}) < \infty, \qquad 1 \leq i \leq n. \tag{3}$$

Observe that since the holding cost rate may be bounded, Assumption 0.3 may apply while some of the queues are unstable.

It can be shown, by similar methods as in [8], [10] and [12], that under Assumption 0.3, there exists a stationary optimal routing policy. That is, the optimal routing decision at time t, depends only on the queue lengths at time t, but not on t.

In this study we find a lower bound of the optimal cost. We first express the lower bound by a solution to a constrained optimization problem. We then explicitly derive the functions that compose the objective function of the optimization problem. These functions are obtained by finding the optimal acceptance policy in a flow control problem of a a single server. Having done so, we then solve the constrained optimization problem from which we construct a class of routing policies, the *overflow routing class.* Finally the cost of an arbitrary policy in this class is analytically evaluated, and numerical examples are computed and compared to the lower bound.

References

[1] Z. Rosberg and P. Kermani. *Customer routing to different servers with complete information.* IBM Research Report, June 1988.

[2] A. D. Bovopoulos and A. A. Lazar. *Optimal Routing and Flow Control of a Network of Parallel Processors.* Technical Report, Columbia University, 1985.

[3] U. Chandy, K. M. Herzog and L. Woo. Parametric analysis of queueing networks. *IBM J. Res. Develop.*, 19:36–42, 1975.

[4] P. Ephremides, A. Varaiya and J. Walrand. A simple dynamic routing problem. *IEEE Trans. on Auto. Contr.*, AC-25:690–693, 1980.

[5] R. Gelenbe and I. Mitrani. *Analysis and Synthesis of Computer Systems.* Academic Press, New York, N.Y., 1980.

[6] B. Hajek. External splitting of point processes. *Math. Operat. Res.*, 10:543–556, 1985.

[7] B. Hajek. Optimal control of two interacting service stations. *IEEE Trans. on Auto. Contr.*, AC-29:491–499, 1984.

[8] S. A. Lippman. Semi-markov decision processes with unbounded rewards. *Management Science*, 19:717–731, 1973.

[9] Z. Rosberg. Deterministic routing to buffered channels. *IEEE Trans. on Communication*, COM-34:504–507, 1986.

[10] Z. Rosberg. A note on the ergodicity of markov chains. *J. Appl. Prob.*, 18:112–121, 1981.

[11] Z. Rosberg and D. Towsley. Customer routing to parallel servers with different rates. *IEEE Trans. on Auto. Contr.*, AC-30:1140–1143, 1985.

[12] M. Schal. Condition for optimality in dynamic programming and for the limit of n-stage optimal policies to be optimal. *Z. Warscheinlichkeitstheorie Verw. Gebiete*, 32:179–196, 1975.

[13] R. R. Weber. On the optimal assignment of customers to parallel servers. *Journal of Appl. Prob.*, 15:406–413, 1978.

[14] W. Whitt. Decoding which queue to join: some counterexamples. *Operat. Res.*, 34:55–62, 1986.

[15] W. L. Winston. Optimality of the shortest line discipline. *Journal of Appl. Prob.*, 14:181–189, 1977.

AVERAGE COST SEMI-MARKOV DECISION PROCESSES AND THE CONTROL OF QUEUEING SYSTEMS

LINN I. SENNOTT
Mathematics Department
Illinois State University
Normal, IL 61761

The mathematical structure underlying the control of many queueing systems is a semi-Markov decision process (SMDP). In this talk, we deal with infinite state SMDPs with finite action sets and unbounded, nonnegative costs. In the final section of the talk, we extend our results to compact action spaces.

A policy is any physically realizable rule for making decisions and may depend on the history of the process. A stationary policy always makes the same decision in a given state and thus is easier to implement. We are interested in optimizing the limiting or long-run expected average cost, over all realizable policies. Assumptions are given under which there exists an expected average cost optimal stationary policy. These conditions generalize our earlier work on MDPs [1]. Readily verifiable sufficient conditions for the assumptions to hold are given.

To illustrate the power of this approach, the existence result is applied to eight queueing control problems. These are: (1) the M/G/1 queue with variable service parameter; (2) the M/G/1 queue with variable service parameter, server shut-down option, and cost of changing actions; (3) the M/G/1 batch processing queue; (4) the Klimov time-sharing network of M/G/1 queues; (5) the G/M/m queue with variable arrival parameter; (6) the G/M/m queue with customer reject option; (7) optimal routing of a Poisson arrival stream to parallel exponential servers, and finally; (8) the M/M/1 queue with service rate in a closed interval.

We briefly state some of the conclusions, omitting a few minor assumptions. An important quantity is $C(i,a)$, the expected undiscounted cost of a transition, given the state is i and the decision a is chosen. Then $C(i,a) = D(i,a) + d(i,a)\tau(i,a)$, where $D(i,a)$ is the instantaneous cost, $d(i,a)$ the cost rate, and $\tau(i,a)$ the average time spent in state i before the next transition (decision point). The quantity $d(i) = \min_a \{d(i,a)\}$ is the minimum cost rate in state i.

For (1) we focus on the number of customers left behind just after the departure of a customer. If $i \geq 1$ customers are left behind, there is a finite set of service parameters a from which the server may choose. It is shown that if $\lim_{i\to\infty} d(i) = \infty$, and if there exists a single service parameter a under which the queue is ergodic, for which the $(n+1)$st service time moment is finite, and such that $C(i,a)$ is bounded above by a polynomial in i of degree n, then there exists an expected average cost optimal stationary policy.

For (3) we again focus on the number of customers left behind just after the completion of a service. If $i \geq 1$, we may choose to service any number k of customers during the next service, where $0 \leq k \leq \min\{i,Q\}$. In any case, the service time distribution does not vary. Assume $\lambda\tau < Q$, where λ is the arrival rate, and τ the mean service time. Assume $\lim_{i\to\infty} d(i) = \infty$. Now let f be the stationary policy that serves $\min\{i,Q\}$ in state i. If $C(i,f(i))$ is bounded above by a polynomial in i of degree n, and if the service time has finite (n+1)st moment, then there exists an expected average cost optimal stationary policy.

As a final example, we discuss (8). In this case, at each decision epoch, the server may choose to serve at exponential rate $\mu \varepsilon [\mu_1, \mu_2]$. Assume that for $i \geq 1$, both $D(i,\mu)$ and $d(i,\mu)$ are lower semicontinuous functions on $[\mu_1, \mu_2]$, and that $\lim_{i \to \infty} d(i) = \infty$. If there exists a service rate $\mu \varepsilon [\mu_1, \mu_2]$ under which the queue is ergodic and for which $C(i,\mu)$ is bounded above by a polynomial in i of degree n, then there exists an expected average cost optimal stationary policy.

These results will appear in [2].

References

[1] L. I. Sennott, Average Cost Optimal Stationary Policies in Infinite State Markov Decision Processes with Unbounded Costs, to appear, Oper. Res.

[2] L. I. Sennott, Average Cost Semi-Markov Decision Processes and the Control of Queueing Systems, to appear, Prob. Eng. and Infor. Sci.

QUICKER CONVERGENCE FOR OPTIMAL STOPPING PROBLEMS THROUGH PROBABILISTIC INTERPRETATIONS OF CLASSICAL NUMERICAL METHODS

ALBERT GREENBERG AND ROBERT VANDERBEI

AT&T Bell Laboratories, Murray Hill NJ 07974

Abstract. Simple probabilistic interpretations of classical numerical methods for solving optimal stopping problems are presented. In typical queueing theory and probabilistic modeling applications the rate of convergence of the computation can be significantly improved by carefully ordering the sequence of updates performed in the course of the computation. Our probabilistic interpretation leads to a simple heuristic to compute an ordering that quickens convergence. Simple efficient asynchronous parallel methods are presented, which retain the benefit of a good ordering while speeding up convergence by a factor of P if P processors participate. The ordering heuristic and parallel methods apply equally well to the problem of computing the invariant measure of a Markov chain.

1. Introduction.

The *optimal stopping problem* involves a Markov chain X_t on a finite state space E, and a nonnegative real valued pay-off function f defined on E. X_t starts at some state $x \in E$. A controller eventually stops X_t at some time τ and collects the pay-off $f(X_\tau)$. We consider the problem of finding the expected pay-off under an optimal control:

$$v(x) = \mathbf{E}_x f(X_{\tau^*}) = \sup_\tau \mathbf{E}_x f(X_\tau),$$

where τ is a stopping time. The function v is called the *value function.* Several types of stochastic problems can be formulated and efficiently solved as optimal stopping problems. For example, there is the problem of finding the probability that a Markov chain exits a given domain through a given set. More generally, there is the Dirichlet problem associated with a Markov chain and a domain in its state space. Several other problems are closely related, such as the problem of finding the expected time a Markov chain needs to exit a given domain.

In this paper we consider the method of successive approximation for computing the value function. In brief, there is an operator that takes an estimate of the value function as input and returns an improved estimate as output. In the parlance of numerical analysis, there are two ways to manage this computation: the *Jacobi* method and the *Gauss-Seidel* method. The Jacobi method keeps two vectors: an old and a new. The old vector is used to calculate a new vector using the successive approximation scheme. Once the new vector is completely determined, it becomes the old vector for the next iteration. The Gauss-Seidel method keeps only one vector, a current estimate. As the successive approximation calculation takes place, the new values are written right back into the same vector. For optimal stopping, both methods work, though the Gauss-Seidel method is simpler and, as we shall see, faster, sometimes considerably. Results of this type are not new. For example, when solving systems of linear equations, if the iteration matrix that appears in the Jacobi scheme is a strict contraction, then Gauss-Seidel is asymptotically faster than Jacobi. This is the well-known Stein-Rosenberg theorem (see e.g. [11] p. 42-44).

Our contributions are as follows. First, simple probabilistic interpretations of the Jacobi and Gauss-Seidel optimal stopping computations are presented (section 2), which give insight into their rates of convergence. A similar interpretation for classical Dirichlet problems (the discrete Laplacian on a square domain) was found independently by Goodman and Madras [3]. Mitra and Tsoucas [9] gave a similar interpretation for the problem of computing the invariant measure of a Markov chain. In typical queueing theory and probabilistic modeling applications the rate of convergence of the Gauss-Seidel method can be significantly improved by carefully ordering the sequence of updates performed in the course of the computation (cf. [12] Appendix D, and [9]). [The opposite is true in applications of the Gauss-Seidel method for some classical highly symmetric problems [3], [13] (cf.

section 2). For those problems, the Gauss-Seidel method converges at twice the speed of the Jacobi method irrespective of the ordering.] Our second contribution is a simple heuristic, termed the *flow deficit method*, to compute an ordering that speeds convergence (section 3). Third, we present two efficient asynchronous parallel Gauss-Seidel algorithms. The algorithms are simple, robust, and easy to implement on a variety of parallel computers. Experiments on a real parallel computer (section 4) suggest that the better of the two methods retains the benefit of a good ordering, while speeding up convergence by factor of P if P processors are put on the computation.

Our heuristic for ordering the Gauss-Seidel method and our parallel methods also apply to the important problem of computing the invariant measure of a Markov chain.

2. Probabilistic Interpretations.

The value function $v(x)$ denotes the expected pay-off if X_t is stopped optimally. When at state x, we could stop X_t immediately and obtain pay-off $f(x)$. On the other hand, we could decide to take one step and then proceed optimally from there, to obtain pay-off $Pv(x)$. By the principle of dynamic programming [2], these two possibilities are sufficient. That is, $v(x)$ is the smallest solution to the following non-linear system of equations:

$$v(x) = \max\{f(x), Pv(x)\}.$$

The optimal strategy is described in terms of the support set of v:

$$\Gamma = \{x : v(x) = f(x)\}.$$

Indeed, τ^* is the first hitting time of Γ:

$$\tau^* = \inf\{t : X_t \in \Gamma\}.$$

In the Jacobi method a sequence of estimates of the value function $v(x)$ are computed:

$$\begin{aligned} v_0(x) &= f(x) && (2.1)\\ v_t(x) &= \max\{v_{t-1}(x), Pv_{t-1}(x)\}, \quad t = 1, 2, 3, \ldots. && (2.2) \end{aligned}$$

It follows from the analysis below (or [2]) that $v_t(x)$ increases monotonically to $v(x)$ as t tends to infinity. To implement the Jacobi method, two copies of the vector v must be kept in store, an old and a new copy. The new copy is calculated using the old copy. When the calculation is complete, the new copy becomes the old copy for the next iteration.

In the Gauss-Seidel method just one vector is held in store. An order relation "$<$" is imposed on the state space E. In general, this relation has *no bearing* on the original problem. An iteration consists of a pass over the states x following the ordering, computing new estimates of $v(x)$, and overwriting the single vector with those estimates. More precisely, given an order relation "$<$" on E, split the transition matrix P into its strict lower part and the remaining upper part: for any function h defined on E,

$$\begin{aligned} Lh(x) &= \sum_{y<x} p(x,y)h(y), && (2.3)\\ Uh(x) &= \sum_{y\geq x} p(x,y)h(y). && (2.4) \end{aligned}$$

The Gauss-Seidel method proceeds as follows:

$$\begin{aligned} w_0(x) &= f(x) && (2.5)\\ w_t(x) &= \max\{w_{t-1}(x), Lw_t(x) + Uw_{t-1}(x)\}. && (2.6) \end{aligned}$$

It follows from Theorems 2.1 and 2.2 below that

$$v_t(x) \leq w_t(x) \leq v(x).$$

Hence, for optimal stopping, the Gauss-Seidel method is always better than the Jacobi method, both in speed of convergence and in storage requirement.

Now, we give the two methods probabilistic interpretations. First, recall that

$$v(x) = \sup_{\tau} \mathbf{E}_x f(X_\tau).$$

Theorem 2.1 $v_t(x) = \sup_{\tau \le t} \mathbf{E}_x f(X_\tau)$.

Let

$$\begin{aligned} \sigma_t &= \inf\{s : s > \sigma_{t-1}, X_s \ge X_{s-1}\} && (2.7) \\ \sigma_0 &= 0. && (2.8) \end{aligned}$$

In words, σ_t represents the time just after the t^{th} high step. In terms of σ_t, we have a result similar to Theorem 2.1 for the t^{th} Gauss-Seidel iterate:

Theorem 2.2 $w_t(x) = \sup_{\tau \le \sigma_t} \mathbf{E}_x f(X_\tau)$.

The most important property of σ_t is that

$$\sigma_t \ge t.$$

Theorem 2.1 says the Jacobi iterate v_t is itself the value function for a finite horizon counterpart to the original optimal stopping problem where the Markov process must stop no later than time t. Similarly, Theorem 2.2 says that the Gauss-Seidel iterate w_t is the value function for a finite horizon counterpart where the Markov process must stop no later than the random time σ_t. Since $\sigma_t \ge t$, these two theorems immediately establish that v_t and w_t are monotonically increasing with $v_t \le w_t$ and limit v. For the proofs of these two theorems, see [6].

Theorem 2.2 shows that the Gauss-Seidel method for calculating the value function converges fastest if the Markov process X_t is likely to take low steps, so $\sigma_{t+1} - \sigma_t$ is likely to be large. For some problems, all orderings are about the same in that $\sigma_{t+1} - \sigma_t$ is nearly constant independent of the ordering. On the other hand, there are natural problems, in particular from queueing theory, where the ordering matters.

Example. Queues in Series. A full infinite buffer of jobs at station 0 feeds station 1, which feeds into station 2, which then drains into an empty infinite buffer. Station i has the capacity to buffer up to N_i jobs, $i = 1, 2$. Service times at station i, $i = 0, 1, 2$, are independent and exponentially distributed with mean $1/\mu_i$. Let n_i denote the number of jobs in station i. Station i starts to service a job if and only if a job is present in its buffer ($n_i > 0$) and the next buffer is not full ($n_{i+1} < N_{i+1}$).

The state of the system (n_1, n_2) evolves as a continuous time Markov chain. From state (n_1, n_2) there are 3 possible transitions:

$$\begin{array}{lll} (n_1 + 1, n_2) & \text{at rate } \mu_0 & \text{if } n_1 < N_1 \\ (n_1 - 1, n_2 + 1) & \text{at rate } \mu_1 & \text{if } n_1 > 0 \text{ and } n_2 < N_2 \\ (n_1, n_2 - 1) & \text{at rate } \mu_2 & \text{if } n_1 > 0 \end{array}$$

If we start the system with both buffers partially loaded, we might want to know the probability that the inventory reaches a given level L, $n_1 + n_2 = L$, before the system reaches another level L', $n_1 + n_2 = L'$. As mentioned in the introduction, this can be formulated as an optimal stopping problem. It suffices to consider the discrete time Markov chain embedded at the times where the state changes, and altered so that states H_L and $H_{L'}$ are absorbing, where $H_K = \{(n_1, n_2) : n_1 + n_2 = K\}$. The pay-off function is one on H_L and zero elsewhere. Hence the value function at (n_1, n_2) is just the probability that the system first hits H_L rather than $H_{L'}$, given that the system starts at (n_1, n_2).

For the sake of discussion, suppose that all station service rates μ_i are equal, the case studied in [9]. Of the large number of choices of orders one might impose on the state space, the sweeping ones are most natural. These are the ones that sweep across successive rows (or columns). We must decide whether to sweep rows then columns or columns then rows. Once this is decided, we must then specify whether we sweep from low to high or from high to low. Hence there are a total of eight sweeping orderings. Each of the eight orderings can be put into one of two categories. First there are those for which two of the three transitions from each state are low steps and those for which two of the three are high steps. The *lexicographic ordering* $(0,0) < (0,1) < \ldots < (0,N_2) < (1,0) < (1,1) < \ldots < (1,N_2) < \ldots$ is in the first category, and the *anti-lexicographic ordering* $(0,0) < (1,0) < \ldots < (N_1,0) < (0,1) < (1,1) < \ldots < (N_1,1) < \ldots$ is in the second. Our probabilistic interpretation suggests that if the density of transitions corresponding to low steps is d then the Gauss-Seidel method should converge at rate $1/(1-d)$ times that of the Jacobi method. Thus, under the lexicographic ordering the Gauss-Seidel computation should need half as many iterations to converge as needed under the anti-lexicographic ordering, and two thirds as many iterations as needed under an ordering chosen uniformly at random. This has been verified experimentally. In the two sections to follow we report the results of several numerical experiments in solving this example.

3. A Heuristic for Ordering the State Space.

In this section we present a heuristic for generating a state space ordering under which the Markov process is likely to take strict low steps. Such orderings quicken the convergence of the Gauss-Seidel method for calculating the value function in optimal stopping.

The method also applies to the Gauss-Seidel method for calculating the invariant measure of a Markov chain. Mitra and Tsoucas [9] showed that orderings under which the chain is likely to take strict high steps quicken convergence. It is trivial to modify our heuristic to generate orderings where strict high steps are likely. Mitra and Tsoucas [9] also gave a sufficient condition to guarantee convergence of the method. Another simple modification to flow deficit gives orderings that (i) make strict high steps likely and (ii) satisfy the Mitra and Tsoucas condition for convergence without underrelaxation.

Flow deficit is a greedy algorithm. Initially all states belong to a candidate set C. A numerical *merit* is associated with each state in C. At step i of the algorithm, a state x of maximal merit is selected, and given position i in the ordering. Before going on to step $i+1$, state x is removed from C, perhaps changing the merits of other states of C.

Let $p(x,y)$ denote the numerical weight associated with the transition from state x to state y; if x and y are the states of a continuous (discrete) time Markov chain then $p(x,y)$ is the rate (probability) associated with the transition from x to y. We consider two possible assignments of merits to the states x in C:

- *inflow – outflow*:

$$merit(x) = \sum_{y \in C} p(y,x) - \sum_{y \in C} p(x,y).$$

- *outflow – inflow*:

$$merit(x) = \sum_{y \in C} p(x,y) - \sum_{y \in C} p(y,x).$$

To compute the value function in an optimal stopping problem, we want an ordering under which the associated random walk is likely to take strict low steps (jumps to states lower in the ordering). In this case *inflow – outflow* is the appropriate measure of merit. Intuitively, for a given state x, if the measure *inflow – outflow* is large then state x is locally attracting, and so should be low in the order. To compute the invariant measure of a Markov chain, we want an ordering under which the associated random walk is likely to take strict high steps (jumps to states higher in the ordering) [9]. In that case *outflow – inflow* is the appropriate measure of merit. Intuitively, this puts locally repelling states low in the order.

Consider our example of queues in series. In this case, the flow deficit method encounters many ties. However, no matter how ties are broken, the order generated always has the property that two out of three transitions from interior states are low steps. In particular, ties can be broken in such a way that

$$(N_1, 0) < (N_1 - 1, 0) < \ldots < (0,0) < (N_1, 1) < (N_1 - 1, 1) < \ldots < (0,1) < \ldots$$

is the order generated.

To conclude this section, we report the results of some experiments testing the flow deficit method on the queues in series example. The tests involved the hitting probability calculation (a special case of optimal stopping) described in the previous section and invariant measure calculation.

Of course it is necessary to measure the distance between successive iterates u and v in the computation. We took

$$\max_i |(u_i - v_i)/v_i|$$

as the measure of distance. The iterations were stopped at the first i where the distance between the $(i-1)$-st and i-th iterates was found to be less than 10^{-6}.

Table 1 shows the results for the problem of queues in series with buffer sizes $N_1 = N_2 = 100$, hitting set levels $L = 100$ and $L' = 0$, and

- rates $\mu_0 = 1$, $\mu_1 = 1$, $\mu_2 = 1$, and
- rates $\mu_0 = 4$, $\mu_1 = 1$, $\mu_2 = 4$.

The table shows the number of iterations needed under

- an ordering generated automatically by our code for the flow deficit method (using *inflow* – *outflow* for hitting probability calculation and *outflow* – *inflow* for invariant measure calculation),
- either lexicographic ordering (hitting probability calculation) or anti-lexicographic ordering (invariant measure calculation), and
- an ordering chosen uniformly at random.

For hitting probability calculation, there is little to choose between the efficiency of the flow deficit ordering and the lexicographic ordering for the balanced case where all $\mu_i = 1$. As expected, about 3/2 times as many iterations are required under the random ordering. The corresponding results for invariant measure calculation are similar. The lexicographic and anti-lexicographic orderings were designed for the balanced case. These orderings do not do well in the unbalanced case where $\mu_0 = \mu_2 = 4$ and $\mu_1 = 1$. This illustrates one difficulty with designing orderings by hand: an ordering good for one set of parameters is not necessarily good for another.

4. Parallel Gauss-Seidel.

In this section we present two simple parallel Gauss-Seidel algorithms. The algorithms are asynchronous, meaning the processors need not run in lockstep. In one of the algorithms, *Async 1*, there is essentially no coordination between processors. In the other, *Async 2*, a mild type of coordination is used, but is used infrequently, and the attendant overhead is negligible.

Both algorithms are easy to implement on a variety of parallel computers. The number of processors dedicated to the computation may vary, even while the computation is in progress. The processors may run at different speeds and may experience variable delays, say, in accessing memory. A few slow processors will not slow down the whole computation, as they would in a synchronous algorithm.

We implemented the algorithms on a Sequent Balance 21000 multiprocessor. All processors share a single, large memory and other resources. For each of our experiments, we reserved a fixed number P of processors, $1 \leq P \leq 16$. While our experiments ran, other jobs that we had no control over ran on different processors.

In the sequential Gauss-Seidel method one processor cycles through the state space, following the fixed ordering. Suppose the total number of states is $M = |E|$ and the states are ordered 0, 1,

Hitting Probability Calculations								
parameters						number of iterations		
μ_0	μ_1	μ_2	N	L	L'	flow deficit	lex.	random
1	1	1	100	100	0	10907	10941	15264
4	1	4	100	100	0	85	184	224

Invariant Measure Calculations						
parameters				number of iterations		
μ_0	μ_1	μ_2	N	flow deficit	anti-lex.	random
1	1	1	100	5716	5858	8229
4	1	4	100	322	408	521

Table 1: Number of Gauss-Seidel iterations needed for the problem of two queues in series with both buffer sizes N.

..., $M-1$. When at the i-th state the processor updates the associated value estimate, and then goes on to state $(i+1) \pmod{M}$.

Our first algorithm, *Async 1*, is the natural generalization for parallel processing. Suppose P processors are available, numbered 0, 1, ..., $P-1$. The basic task of processor p is the same as in the sequential algorithm: Sweep through the state space following the ordering and updating the associated values. However, instead of starting its sweep at the state 0, processor p starts at state g_p, where g_p is a parameter, and updates states in the order g_p, g_p+1, ..., $M-1$, 0, 1, ..., g_p-1.

How should the g_p be chosen? A snapshot of the computation would reveal a subset of *active* states $A \subset \{0,1,...,M\}$ of size $|A| \leq P$, the number of processors. ($|A| < P$ happens if two or more processors simultaneously update the same state.) If the gaps between the active states are small, then the processors will be bunched together during their sweeps. As a result:

- Not infrequently, two processors will attempt to update the same state simultaneously, an obvious waste.
- Since typically the ordering is such that value computed at state p depends strongly on the values of nearby states, simultaneous updates will be made on the basis of similar data. Hence the computation will behave like the inferior Jacobi method.

Thus, it appears that the gaps between the g_p should be large. In our experiments, we took

$$g_p = p\lfloor \frac{M}{P} \rfloor,$$

so the gaps are as large as possible. Though this choice initially separates the active states, it gives no guarantee that the states remain separated throughout the computation.

Our second algorithm, *Async 2*, uses a variant of the *fetch_and_add* instruction [4], [5], [10] to regulate the separation. We need

$$a \leftarrow mod\_fetch\_and\_add(x,b,c),$$

to be equivalent to an indivisible execution of the two instructions

$$\begin{aligned} a &\leftarrow x \\ x &\leftarrow x+b \pmod{c}, \end{aligned}$$

where a is a private variable, and x is a shared variable. We implemented *mod_fetch_and_add* on the Sequent multiprocessor using a hardware locking mechanism.

Under *Async 2*, each processor repeatedly uses *mod_fetch_and_add* to obtain a pointer to a large interval of states, and then performs a Gauss-Seidel sweep of the interval. Though two processors may approach one another, nearly updating the same states at the same time, the two separate after executing the *mod_fetch_and_add* to obtain pointers to their next intervals. The details are as follows. A shared variable *next_interval* is initialized to 0. Each processor repeatedly executes

$$my\_interval \leftarrow mod\_fetch\_and\_add(next\_interval, K, M),$$

where *my_interval* is a private variable and K is a parameter, and performs a Gauss-Seidel sweep of the K states $my\_interval$, $(my\_interval + 1) \pmod M$, ..., $(my\_interval + K - 1) \pmod M$. We took $K = \max\{1, \lfloor M/P \rfloor\}$ in our experiments.

For each sequence of M individual updates, a processor computes an aggregate error estimate

$$\max_i |(u_i - v_i)/v_i|,$$

where v_i represents the new value computed at the processors i-th update and u_i the old value. Informally, when a processor finds the aggregate error less than a given ϵ it raises its hand (increments a counter using *fetch_and_add*), and when it finds the aggregate error greater than or equal to ϵ it lowers its hand (decrements the counter). After computing M updates a processor goes on to compute another M unless all processor have their hands raised (the counter equals P, the total number of processors).

For both *Async 1* and *Async 2*, this rule ensures that in a parallel pass over the whole state space, the most recent update at each state resulted in a uniformly small change. In other words, the error computed over a Gauss-Seidel sweep of all states, possibly visiting some states more than once, is uniformly small. We mention a small variation on this stopping rule, which we also tried. After detecting termination as just described, one processor makes an additional sweep over the state space, computing the error but not storing the updates. If this error, which we call the Jacobi residual, is less than ϵ then the computation is simply restarted. This continues until the Jacobi residual is found to be less than ϵ. In our experiments, in all cases just one pass was required: On the first check the Jacobi residual was always found to be less than ϵ.

To test the efficiency of *Async 1* and *Async 2*, we ran experiments similar to those described in the previous section for calculating invariant measures and hitting probabilities for the queues in series example. We shall report just the results for the hitting probability calculation; the results for invariant measure calculation were analogous. We set buffer sizes $N_1 = N_2 = 50$, hitting set levels $L = 50$ and $L' = 0$, and rates $\mu_0 = \mu_1 = \mu_2 = 1$. The flow deficit method was used to generate the state space ordering. The parameter ϵ of the stopping rule was taken to be 10^{-6}. This is a moderate size problem. There are 2500 states. On one processor, 3303 iterations are required, where an iteration is a sweep of the state space, consisting of M individual updates.

We assessed the efficiency of the parallel computation as follows. For $p = 0, 1, ..., P - 1$, let $T_P(p)$ denote the number of iterations that processor p performs, where an iteration consists of M individual updates. (Thus, in *Async 2*, an iteration is M/K sweeps of intervals of size K.) Let

$$T_P = \max_{0 \le p \le P-1} T_P(p).$$

As a measure of efficiency we took the *speedup*, defined as

$$S(P) = T_1/T_P,$$

the ratio of the time to solve the problem with one processor to the time to solve the problem with P processors.

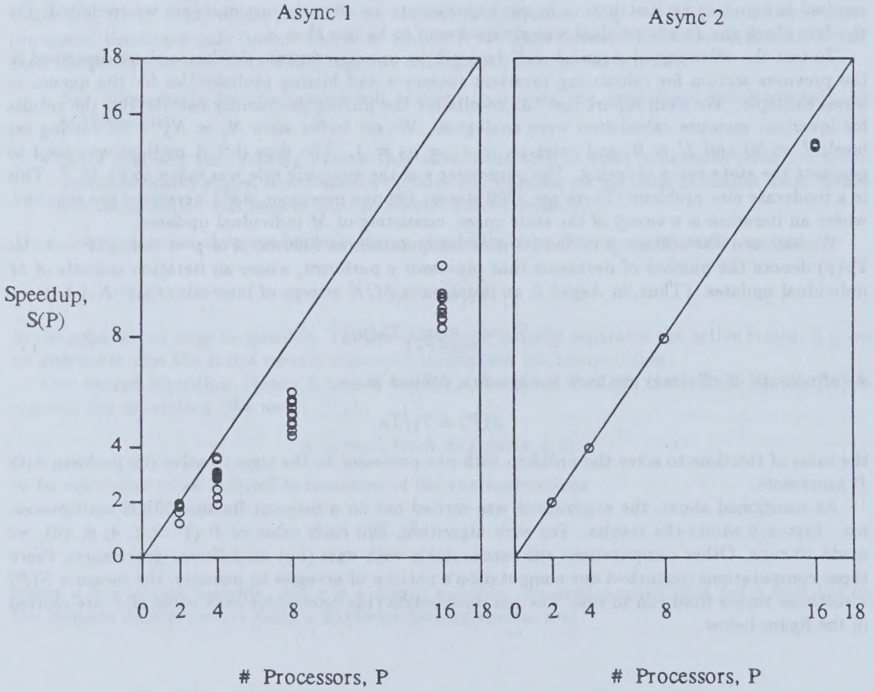
As mentioned above, the experiment was carried out on a Sequent Balance 21000 multiprocessor. Figure 6 shows the results. For each algorithm, and each value of P ($P = 2, 4, 8, 16$), we made 10 runs. Other computations ran concurrently with ours (but on different processors). Since those computations perturbed our computation's pattern of accesses to memory, the measure $S(P)$ sometimes varied from run to run. The ten data points (the circles) for each value of P are plotted in the figure below.

Under *Async 1* the spread in the results from run to run is significant. For example, for $P = 16$ there is a value of $S(P)$ about 50% larger than the smallest value. Under *Async 2* the spread in the results is imperceptible, and $S(P)$ is about P. (In all cases $S(P) \leq P$ though this is hard to discern from the Figure.) It appears hard to do better. (It might be possible to do better, since serialization of the updates performed in the parallel cases could give an ordering that in the sequential case leads to faster convergence than the ordering generated by the flow deficit method.)

Async 1 sometimes performs as well as *Async 2*, but the penalty for its higher degree of asynchrony is that it sometimes performs significantly worse. As P, the number of processors approaches M, the number of states, the algorithms behave more and more like the Jacobi method, and the slope of the speedup decreases. We note that on a synchronous parallel processor dedicated to the Gauss-Seidel computation there would be little difference between the two algorithms.

Some final remarks are in order. On first blush, simpler ways to separate the active states are appealing. The parallel task queue idea of [4] could be used. However, the coordination overhead would be greater than under *Async 2*. Even simpler, split the M states into P intervals each of size about M/P. For each interval, permanently assign one processor to perform Gauss-Seidel sweeps. However, this algorithm appears to run at the speed of the slowest processor. In contrast, both *Async 1* and *Async 2* appear to be hardly effected by a few slow processors. For example, we ran the experiment described above with $P = 16$, but with two processors artificially slowed to half their normal speed. (A dummy job ran on the two processors time-shared with the Gauss-Seidel job.) The difference between the resulting speedup data and that plotted in Figure 6 was quite small, and consistent with the view that the effective number of processors was 15 rather than 16.

Acknowledgment. We thank Boris Lubachevsky and Debasis Mitra for very helpful discussions about their work. We thank Dan Heyman for his questions about provably convergent orderings for the invariant measure calculations.



References

[1] Chazan, D. and Miranker, W., "Chaotic Relaxation". *Lin. Alg. and Its Applications,* 2 (1969), 199-222.

[2] Dynkin, E.B. and Yushkevich, A.A., *Markov Processes, Theorems and Problems,* 1969, Plenum Press, New York.

[3] Goodman, J. and Madras, N., "Random walk interpretations of classical iteration methods", preprint (1987).

[4] Gottlieb, A., Lubachevsky, B.D., and Rudolph, L., "Basic techniques for the efficient coordination of very large numbers of cooperating sequential processors," *ACM Transactions on Programming Languages and Systems,* 5, 2 (April 1983), 164-189.

[5] Gottlieb, A., Grishman, R., Kruskal, C.P., McAuliffe, K.P., Rudolph, L., and Snir, M., "The NYU ultracomputer - designing an MIMD shared memory parallel machine," *IEEE Transactions on Computers,* C-32, 2 (February 1983).

[6] Greenberg, A.G., and Vanderbei, R.J., "Quicker convergence for optimal stopping problems through probabilistic interpretations of classical numerical methods," preprint (1988).

[7] Lubachevsky, B.D., "An approach to automating the verification of compact parallel coordination programs I," *Acta Inf.,* 21 (1984), 125-169.

[8] Lubachevsky, B.D. and Mitra, D., "A chaotic asynchronous algorithm for computing the fixed point of a nonnegative matrix of unit spectral radius", *J. ACM,* vol. 33, no. 1 (January, 1986), 130-150.

[9] Mitra, D. and Tsoucas, P., "Convergence of relaxations for numerical solutions of stochastic problems", *Mathematical Computer Performance and Reliability,* (Proceedings of the Second International Workshop on Applied Mathematics and Performance/Reliability Models of Computer/Communications Systems, University of Rome II, May 1987, pp. 119-133), North Holland, 1987.

[10] Rudolph, L., *Software Structures for Ultraparallel Computing,* Ph.D. Thesis, Courant Institute, New York University (December 1981).

[11] Seneta, E., *Non-negative Matrices and Markov Chains, 2nd. Ed.,* 1981, Springer-Verlag, New York.

[12] Tijms, H.C., *Stochastic Modelling and Analysis, a Computational Approach* 1986, John Wiley and Sons, New York.

[13] Varga, R., *Matrix Iterative Analysis,* 1962, Prentice Hall, Englewood Cliffs, NJ.

ENTROPY RATE OF STATIONARY RANDOM FIELDS ON BINARY TREE

TOBY BERGER
School of Electrical Engineering and
Center for Applied Mathematics
Cornell University
Ithaca, NY 14853 USA

ZHONGXING YE
Center for Applied Mathematics
Cornell University
Ithaca, NY 14853
U.S.A.

I. ABSTRACT

We show that the entropy rate exists for any stationary random field on a binary tree. Three proofs are given, each of which is of interest in its own right. The first is a direct and simple proof. The second is based on the fact that the entropy rates on finite subtrees are nonincreasing. The third employs an alternate representation of entropy rate in term of the conditional entropy rate.

II. DEFINITIONS AND NOTATIONS

A tree is a graph $G = \{T, E\}$ which is connected and contains no circuits. A binary tree T_b is a tree possesses a root vertex $\{0\}$ and is such that each vertex has two branches to the next level (see Figure 1). A vertex is said to be at n-th level if the path linking it and the root contains n edges. We use (n, j) to denote the j-th vertex at the n-th level.

A random field on T_b is a set of random variables $\mathbf{X} = \{X_t,\ t \in T_b\}$. Each X_t takes value over a finite or infinite set S. A random field on T_b is called stationary if the probability distribution of all finite cylinder sets remain invariant under any shift operation, both when each vertex is replaced by its left descendent and when each vertex is replaced by its right descendent. More definitions about the random fields of tree can be found in [1]-[3].

---

[1] This work was partially supported by the U.S. Army Research Office through the Mathematical Sciences Institute at Cornell University and by the National Science Foundation under grant ECS-8305681.

Consider a random field on binary tree $\mathbf{X} = \{X_j, j \in T_b\}$. We use $T^{(n)}$ to denote the finite subtree with n levels and then define the following entropic quantities

Definition 1 Entropy rate: We let

$$h_n(\mathbf{X}) = \mid T^{(n)} \mid^{-1} H(X^{T^{(n)}}) \tag{1}$$

and then define the entropy rate by

$$h(\mathbf{X}) = \lim_{n\to\infty} h_n(\mathbf{X}) \tag{2}$$

if the limit exists.

Definition 2 Conditional entropy rate: We denote n-th level sites by L_n, the n-th level random vector by

$$X^{L_n} = (X_{n,1}, X_{n,2}, \cdots, X_{n,2^n}),$$

and the n-th level conditional entropy rate by

$$h_n^L(\mathbf{X}) = \frac{1}{\mid L_n \mid} H(X^{L_n} \mid X^{T^{(n-1)}}) = \frac{1}{2^n} H(X^{L_n} \mid X^{T^{(n-1)}}). \tag{3}$$

Then the conditional entropy rate is defined by

$$h^L(\mathbf{X}) = \lim_{n\to\infty} h^L(\mathbf{X}) \tag{4}$$

III. MAIN RESULTS: THREE THEOREMS

We establish the following three theorems via information-theoretic arguments. The foundamental properties of information quantities that we use can be found in [4].

Theorem 1 For any stationary random field on the binary tree T_b, the entropy rate $h(\mathbf{X})$ exists.

Proof:

We decompose the finite tree $T^{(n)}$ into three parts $T_1^{(n-1)}$, $T_2^{(n-1)}$ and the root $\{0\}$ (see Figure 1). Then

$$\begin{aligned}
H(X^{T^n}) &= H(X^{T_1^{(n-1)}}) + H(X^{T_2^{(n-1)}} \mid X^{T_1^{(n-1)}}) + H(X_0 \mid X^{T^{(n)}-\{0\}}) \\
&\leq 2H(X^{T^{(n-1)}}) + H(X_0) \\
h_n(\mathbf{X}) &= \frac{1}{2^{n+1}-1} H(X^{T^{(n)}}) \\
&\leq \frac{2^{n+1}-2}{2^{n+1}-1} h_{n-1}(\mathbf{X}) + \frac{1}{2^{n+1}-1} H(X_0) \\
&\leq h_{n-1}(\mathbf{X}) + \frac{1}{2^n} H(X_0)
\end{aligned} \tag{5}$$

Using the above inequality recursively, we get

$$\begin{aligned}
h_{n+1}(\mathbf{X}) &\leq h_n(\mathbf{X}) + \frac{1}{2^{n+1}} H(X_0) \\
h_{n+2}(\mathbf{X}) &\leq h_{n+1}(\mathbf{X}) + \frac{1}{2^{n+2}} H(X_0) \\
&\leq h_n(\mathbf{X}) + (\frac{1}{2^{n+1}} + \frac{1}{2^{n+2}}) H(X_0) \\
\cdots\cdots & \\
h_{n+k}(\mathbf{X}) &\leq h_n(\mathbf{X}) + (\frac{1}{2^{n+1}} + \frac{1}{2^{n+2}} + \cdots + \frac{1}{2^{n+k}}) H(X_0) \\
&\leq h_n(\mathbf{X}) + \frac{1}{2^n} H(X_0) \qquad for\ all\ k = 1, 2, \cdots \qquad (6)
\end{aligned}$$

Then

$$\sup_k h_{n+k}(\mathbf{X}) \leq h_n(\mathbf{X}) + \frac{1}{2^n} H(X_0) \qquad (7)$$

But the sequence $\sup_k h_{n+k}(\mathbf{X})$ is nonincreasing with n. So

$$\inf_n \sup_k h_{n+k}(\mathbf{X}) \leq \inf_n [h_n(\mathbf{X}) + \frac{1}{2^n} H(X_0)] \qquad (8)$$

By taking the limit, we get

$$\overline{\lim_{n\to\infty}} h_n(\mathbf{X}) \leq \underline{\lim}_{n\to\infty} h_n(\mathbf{X}) \qquad (9)$$

The inverse inequality is obvious, so the entropy rate exists.

Theorem 2 For any stationary random field on the tree T_b, $h_n(\mathbf{X})$ is nonnegative and nonincreasing with n increasing. Therefore the entropy rate $h(\mathbf{X})$ exists.

Proof:

First we prove a lemma.

Lemma 1 For the same decomposition of the finite subtree $T^{(n)}$ as in Theorem 1, the mutual informations $I(X^{T_1^{(n)}}; X^{T_2^{(n)}})$ and $I(X_0; X^{T_1^{(n)}} X^{T_2^{(n)}})$ are nondecreasing as n increasing.

Proof:

These are obviously true, because the more random variables we put in, the more mutual information we have. Rigorously we can prove them as follows.

$$\begin{aligned}
I(X^{T_1^{(n-1)}}; X^{T_2^{(n-1)}}) &= H(X^{T_1^{(n-1)}}) - H(X^{T_1^{(n-1)}} \mid X^{T_2^{(n-1)}}) \\
&\leq H(X^{T_1^{(n-1)}}) - H(X^{T_1^{(n-1)}} \mid X^{T_2^{(n)}}) \\
&= I(X^{T_1^{(n-1)}}; X^{T_2^{(n)}}) \\
&= H(X^{T_2^{(n)}}) - H(X^{T_2^{(n)}} \mid X^{T_1^{(n-1)}})
\end{aligned}$$

$$
\begin{aligned}
&\leq H(X^{T_2^{(n)}}) - H(X^{T_2^{(n)}} \mid X^{T_1^{(n)}}) \\
&= I(X^{T_1^{(n)}}; X^{T_2^{(n)}})
\end{aligned}
$$

Similarly we can prove that

$$
I(X_0; X^{T_1^{(n-1)}} X^{T_2^{(n-1)}}) \leq I(X_0; X^{T_1^{(n)}} X^{T_2^{(n)}}). \tag{10}
$$

Now we turn to the proof of the theorem. The nonnegativity is obvious, it suffices to show that $h_n(\mathbf{X})$ is nonincreasing as n increases. We have

$$
\begin{aligned}
H(X^{T^{(n)}}) &= H(X^{T_1^{(n-1)}}) + H(X^{T_2^{(n-1)}} \mid X^{T_1^{(n-1)}}) + H(X_0 \mid X^{T_1^{(n-1)}} X^{T_2^{(n-1)}}) \\
&\quad + H(X^{T_2^{(n-1)}}) - H(X^{T_2^{(n-1)}}) + H(X_0) - H(X_0) \\
&= H(X_0) + 2H(X^{T^{(n-1)}}) - I(X^{T_1^{(n-1)}}; X^{T_2^{(n-1)}}) \\
&\quad - I(X_0; X^{T_1^{(n-1)}} X^{T_2^{(n-1)}})
\end{aligned} \tag{11}
$$

where $H(X^{T_1^{(n-1)}}) = H(X^{T_2^{(n-1)}}) = H(X^{T^{(n-1)}})$ because of the stationarity, and $I(\mathbf{Y}; \mathbf{Z}) = H(\mathbf{Y}) - H(\mathbf{Y} \mid \mathbf{Z})$ is the mutual information of random vectors $\mathbf{Y}$ and $\mathbf{Z}$.

Dividing both sides of (11) by $\mid T^{(n)} \mid = 2^{n+1} - 1$, we get

$$
\begin{aligned}
h_n(\mathbf{X}) &= \frac{2^{(n+1)} - 2}{2^{n+1} - 1} h_{n-1}(\mathbf{X}) + \frac{1}{2^{n+1} - 1} [H(X_0) - I(X^{T_1^{(n-1)}}; X^{T_2^{(n-1)}}) \\
&\quad - I(X_0; X^{T_1^{(n-1)}} X^{T_2^{(n-1)}})]
\end{aligned} \tag{12}
$$

Then

$$
\frac{2^{n+1} - 2}{2^{n+1} - 1} h_n(\mathbf{X}) = \frac{2^{n+1} - 2}{2^{n+1} - 1} h_{n-1}(\mathbf{X}) + \frac{1}{2^{n+1} - 1} \cdot
$$

$$
[H(X_0) - I(X^{T_1^{(n-1)}}; X^{T_2^{(n-1)}}) - I(X_0; X^{T_1^{(n-1)}} X^{T_2(n-1)}) - h_n(\mathbf{X})] \tag{13}
$$

If the term in the bracket $[\cdots]$ is less than 0, the proof will be completed. Actually using (11) recursively that term equals

$$
\begin{aligned}
[\cdots] &= H(X_0) - I(X^{T_1^{(n-1)}} : X^{T_2^{(n-1)}}) - I(X_0; X^{T_1^{(n-1)}} X^{T_2^{(n-1)}}) \\
&\quad - \frac{1}{2^{n+1} - 1} [H(X_0) + 2H(X^{T^{(n-1)}}) - I(X^{T_1^{(n-1)}}; X^{T_2^{(n-1)}}) \\
&\quad - I(X_0; X^{T_1^{(n-1))}} X^{T_2^{(n-1)}})] \\
&= H(X_0) - I(X^{T_1^{(n-1)}}; X^{T_2^{(n-1)}}) - I(X_0; X^{T_1^{(n-1)}} X^{T_2^{(n-1)}}) - \frac{1}{2^{n+1} - 1} \cdot \\
&\quad [H(X_0) - I(X^{T_1^{(n-1)}}; X^{T_2^{(n-1)}}) - I(X_0; X^{T_1^{(n-1)}} X^{T_2^{(n-1)}})
\end{aligned}
$$

$$
\begin{aligned}
& +2H(X_0) - 2I(X^{T_1^{(n-2)}}; X^{T_2^{(n-2)}}) - 2I(X_0; X^{T_1^{(n-2)}} X^{T_2^{(n-2)}}) \\
& +2^2 H(X_0) - 2^2 I(X^{T_1^{(n-3)}}; X^{T_2^{(n-3)}}) - 2^2 I(X_0; X^{T_1^{(n-3)}} X^{T_2^{(n-)}}) \\
& + \cdots \\
& +2^{n-2} H(X_0) - 2^{n-2} I(X^{T_1^{(1)}}; X^{T_2^{(1)}}) - 2^{n-2} I(X_0; X^{T_1^{(1)}} X^{T_2^{(1)}}) \\
& +2^{n-1} H(X_0) - 2^{n-1} I(X_{1,1}; X_{1,2}) - 2^{n-1} I(X_0; X_{1,1} X_{1,2}) \\
& +2^n H(X_0)] \\
= \; & -I(X^{T_1^{(n-1)}}; X^{T_2^{(n-1)}}) - I(X_0; X^{T_1^{(n-1)}} X^{T_2^{(n-1)}}) \\
& + \frac{1 + 2 + 2^2 + \cdots + 2^{n-1}}{2^{n+1} - 1} [I(X^{T_1^{(n-1)}}; X^{T_2^{(n-1)}}) + I(X_0; X^{T_1^{(n-1)}} X^{T_2^{(n-1)}})] \\
= \; & -(1 - \frac{2^n - 1}{2^{n+1} - 1})[I(X^{T_1^{(n-1)}}; X^{T_2^{(n-1)}}) + I(X_0; X^{T_1^{(n-1)}} X^{T_2^{(n-1)}})] \\
\leq \; & -1/2[I(X^{T_1^{(n-1)}}; X^{T_2^{(n-1)}}) + I(X_0; X^{T_1^{(n-1)}} X^{T_2^{(n-1)}})] \leq 0
\end{aligned}
$$

where the last inquality holds because of the nonnegativity of mutual information. Thus the theorem is proved.

Theorem 3 For any stationary random field on T_b, $h_n^L(\mathbf{X})$ is nonnegative and nonincreasing with n. Therefore the limit $h^L(\mathbf{X})$ exists. Moreover, it equals to the entropy rate $h(\mathbf{X})$.

Proof:

We spilit the n-th level random vertor $X^{L_n} = X_{n,1} X_{n,2}, \cdots, X_{n,2^n}$ into two parts (see Figure 2), left haft part

$$X^{L_n^l} = X_{n,1}, \cdots, X_{n,2^{n-1}}$$

and right half part

$$X^{L_n^r} = X_{n,2^{n-1}+1}, \cdots, X_{n,2^n}.$$

Then

$$
\begin{aligned}
H(X_{L_n} \mid X^{T^{(n-1)}}) &= H(X^{L_n^l} \mid X^{T^{(n-1)}}) + H(X^{L_n^r} \mid X^{T^{(n-1)}} X^{L_n^l}) \\
&\leq H(X^{L_n^l} \mid X^{T_1^{(n-2)}}) + H(X^{L_n^r} \mid X^{T_2^{(n-2)}}) \\
&= 2H(X^{L_{n-1}} \mid X^{T^{(n-2)}})
\end{aligned}
$$

Therefore

$$h_n^L(\mathbf{X}) = \frac{1}{2^n} H(X^{L_{n-1}} \mid X^{T^{(n-2)}})$$

$$
\begin{aligned}
&\le \frac{1}{2^{n-1}} H(X^{L_{n-1}} \mid X^{T^{(n-2)}}) \\
&= h^L_{n-1}(\mathbf{X})
\end{aligned}
$$

This, together with the obvious nonnegativity, proves the existence of the limit $h^L(\mathbf{X})$. Then we use the chain formula

$$
H(X^{T^{(n)}}) = H(X_0) + H(X^{L_1} \mid X_0) + H(X^{L_2} \mid X^{T^{(1)}}) + \cdots + H(X^{L_n} \mid X^{T^{(n-1)}})
$$

Dividing both sides by $(2^{n+1} - 1)$, we have

$$
\begin{aligned}
h_n(\mathbf{X}) &= \frac{1}{2^{n+1}-1} H(X^{T^{(n)}}) \\
&= \frac{1}{2^{n+1}-1} h^L_0(\mathbf{X}) + \frac{2}{2^{n+1}-1} h^L_1(\mathbf{X}) + \frac{2^2}{2^{n+1}-1} h^L_2(\mathbf{X}) \\
&\quad + \cdots + \frac{2^n}{2^{n+1}-1} h^L_n(\mathbf{X})
\end{aligned}
$$

Notice two facts: (1) In the right hand side the sum of coefficients is 1 for all $n = 0, 1, 2, \cdots$. (2) The limit $\lim_{n\to\infty} h^L_n(\mathbf{X}) = h^L(\mathbf{X})$ exists. Then by the following lemma in which $S_n = h_n(\mathbf{X}), \quad b_n = h^L_n(\mathbf{X})$, we can get

$$
h(\mathbf{X}) = h^L(\mathbf{X}).
$$

Lemma 2 Assume $\{ \ b_n \ \}_0^\infty$ is a nonnegative and nonincreasing sequence, and $\lim_{n\,\to\,\infty} b_n \ = \ b$,

$$
a_{n,k} = \frac{2^k}{2^{n+1}-1}, \quad k = 0, 1, 2, \cdots, n; \quad n = 0, 1, 2, \cdots.
$$

$$
S_n = a_{n,0} b_0 + a_{n,1} b_1 + \cdots + a_{n,n} b_n
$$

then $\lim_{n\to\,\infty} S_n \ = \ b$.

Proof:

Since $\lim_n b_n = b$, for any $\epsilon > 0$ there exists N such that for all $n > N$,

$$
\mid b_n - b \mid < \epsilon.
$$

For this fixed N,

$$
S_{N+k} = a_{N+k,0} b_0 + \cdots + a_{N+k,N} b_N + a_{N+k,N+1} b_{N+1} + \cdots + a_{N+k,N+k} b_{N+k}
$$

Therefore

$$\begin{aligned}
\mid S_{N+k}-b \mid \;&\leq\; \mid a_{N+k,0}(b_0-b)\mid + \cdots + \mid a_{N+k,N}(b_N-b)\mid \\
&\quad +(a_{N+k,N+1}+\cdots+a_{N+k,N+k})\epsilon \\
&\leq\; \frac{2^{N+1}-1}{2^{N+k+1}-1}\max_{0\leq l\leq N}\mid b_l-b\mid + \epsilon,
\end{aligned}$$

Letting $k \to \infty$, we get

$$\mid \lim_{n\to\infty} S_n - b \mid \leq \epsilon.$$

Because ϵ can be chosen arbitrarily small, we have proved that

$$\lim_n S_n = b.$$

Remark:

Related results can be found in our relative work [5].

Acknowledgment

Professor V.Anantharam suggested the proof of Theorem 1 before we publicly present our proof of Theorem 2 [6], and professor T.Cover suggested the proof of Theorem 3 after hearing that presentation.

References

[1] F.Spitzer, "Markov random fields on an infinite tree," Annals of Prob., vol. 3, No.3, 387-398, 1975.

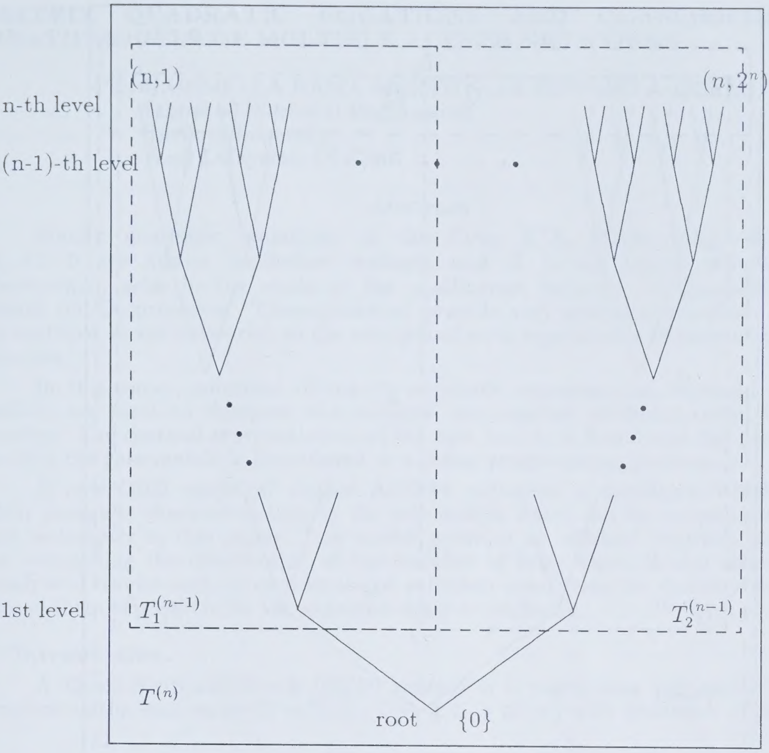
[2] C.J.Preston, Gibbs States on Countable Set, Cambridge University Press, Cambridge, England. 1974.

[3] R.Kindermann, J.L.Snell, Markov random fields and their applications. (contemporary mathematics vol. 1) AMS Providence Rhode Island, 1980.

[4] R.G.Gallager, Information Theory and Reliable Communication. New York, Wiley. 1968.

[5] Zhongxing Ye, "Entropy and ϵ-entropy of random field", doctoral dissertation, Center for Applied Math., Cornell University, Ithaca, New York, 1988. In preparation.

[6] T.Berger, Zhongxing Ye, "The entropy rate of a stationary random field on a tree is monotonic nonincreasing with depth", 1988 IEEE International Symposium on Information Theory, recent result session, Kobe, Japan, June 20, 1988.



n-th level
(n-1)-th level
1st level
(n,1)
$(n, 2^n)$
$T_1^{(n-1)}$
$T_2^{(n-1)}$
$T^{(n)}$
root
$\{0\}$

Figure 1 Decomposition of tree $T^{(n)}$, case 1

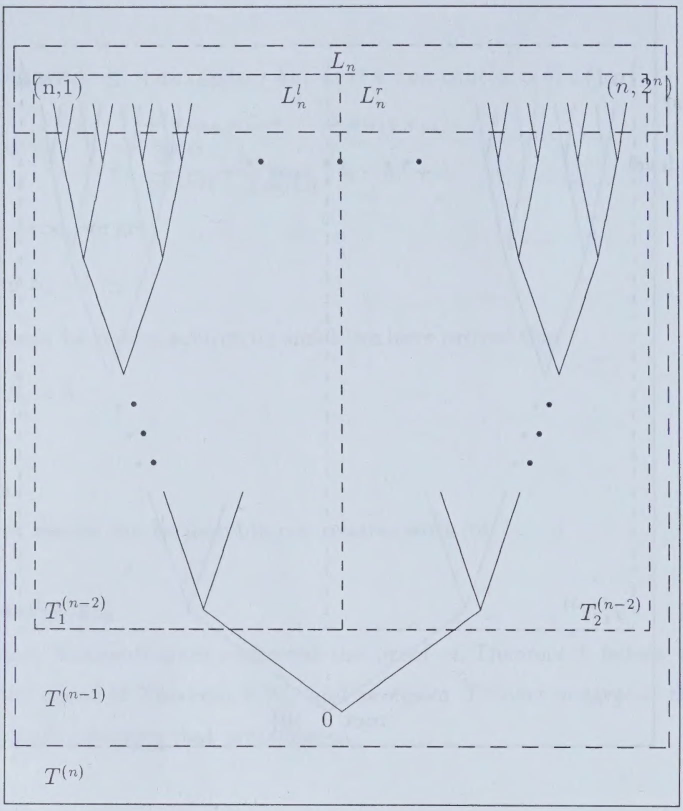

L_n
(n,1)
L_n^l
L_n^r
$(n, 2^n)$
$T_1^{(n-2)}$
$T_2^{(n-2)}$
$T^{(n-1)}$
0
$T^{(n)}$

Figure 2 Decomposition of tree $T^{(n)}$, case 2

MATRIX QUADRATIC EQUATIONS AND QUASI-BIRTH-AND-DEATH MODELS OF MULTIPLE ACCESS NETWORKS

GARIMELLA RAMA MURTHY and EDWARD J. COYLE
School of Electrical Engineering
Purdue University
West Lafayette, IN 47907

Abstract

Matrix quadratic equations of the form $\mathbf{X}^2\mathbf{A}_2 + \mathbf{X}\mathbf{A}_1 + \mathbf{A}_0 = \mathbf{0}$, where $\mathbf{A}_i$, $i \geqq 0$ are known coefficient matrices and $\mathbf{X}$ is the square matrix to be determined, arise in the study of the equilibrium behavior of Quasi-Birth-and-Death (QBD) processes. These processes provide very good models of the activity in multiple access networks, so the solution of such equations is of current practical interest.

In this paper, solutions of matrix quadratic equations are explored and the results are used to compute the minimal nonnegative solution, called the *rate matrix*. The spectral representation of the rate matrix is found and the problem of finding the rate matrix is formulated as a linear programming problem.

A new QBD model of slotted ALOHA networks is developed which allows their complete characterization by the rate matrix which can be computed by using the techniques in this paper. This model provides an efficient recursive algorithm for computing the distribution of the number of busy users. It also allows many analytical results such as necessary and sufficient conditions for stability and exact closed form expression for the expected delay to be found.

1. Introduction.

A Quasi-Birth-and-Death (QBD) process is a continuous parameter Markov process on the state space $E = \{(i,j) : i \geq 0, 1 \leqq j \leqq n\}$ with generator of the form

$$\mathbf{Q} = \begin{bmatrix} \mathbf{B}_0 & \mathbf{A}_0 & \mathbf{0} & \mathbf{0} & \cdots \\ \mathbf{B}_1 & \mathbf{A}_1 & \mathbf{A}_0 & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{A}_2 & \mathbf{A}_1 & \mathbf{A}_0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \end{bmatrix}. \tag{1.1}$$

The formal study of these processes was initiated by Evans [4] and continued by Neuts [3].

QBD processes provide natural models of the activity on multiple access computer networks such as CSMA [6,7]. These processes are also utilized in the analysis of priority queuing system arising in integrated networks [8].

One of the important advantages of the models based on QBD processes is the existence of a recursive solution, called the *matrix geometric solution* for their limiting probabilities. The solution can be described as follows. Define $\overline{\pi}_n$ to be the vector of equilibrium probabilities for the states on level n of the QBD process. Neuts showed in [3] that there exists a matrix $\mathbf{R}$, called the *rate matrix,* such that

$$\overline{\pi}_{n+1} = \overline{\pi}_n \mathbf{R} \qquad n>1. \tag{1.2}$$

The rate matrix $\mathbf{R}$ is the minimal nonnegative solution of the matrix quadratic equation

$$\mathbf{X}^2\mathbf{A}_2 + \mathbf{X}\mathbf{A}_1 + \mathbf{A}_0 = \mathbf{0}, \tag{1.3}$$

where the $\mathbf{A}_i$ matrices are those in the generator in (1.1). Thus the equilibrium analysis of QBD processes essentially depends on the computation of $\mathbf{R}$.

There are primarily two approaches to compute the rate matrix. In the first approach, an explicit expression for **R** is found or it is approximated by iterative procedures. In this direction, previously, an expression for **R** is found only by assuming a special structure of $\mathbf{A}_i$ matrices [8,9]. But the general problem of finding the spectral representation of **R**, the minimal nonnegative solution of a matrix quadratic equation was an unsolved problem. So the iterative procedures were the only resort when the $\mathbf{A}_i$ matrices are arbitrary. The iterative procedures are approximate in nature and often require both large processing times and storage space.

In the second approach, a state space expansion technique is used to find a new matrix **W**, which provides a matrix geometric solution and also satisfies (1.3) [5]. Now, by relating **R** to **W**, the spectral representation of **R** is found. Thus, this approach, though it provides the spectral representation of **R**, requires expansion of the state space and the calculation of **W**.

In this paper, more direct methods of finding the rate matrix are developed. First the general problem of solving a matrix quadratic equation is explored. The techniques are utilized to develop two methods of finding **R**. These results provide an efficient recursive algorithm for computing various performance measures of a QBD model of slotted ALOHA networks. Several analytical results such as necessary and sufficient conditions for stability, the distribution of number of busy users, expected delay and throughput are found.

2. Solutions of Matrix Quadratic Equations.

Consider a matrix quadratic equation of the form

$$\mathbf{X}^2\mathbf{C}_2 + \mathbf{X}\mathbf{C}_1 + \mathbf{C}_0 = \mathbf{0}\,, \tag{2.1}$$

where $\mathbf{C}_0$, $\mathbf{C}_1$, $\mathbf{C}_2$ are known coefficient matrices and **X** is an unknown n×n matrix. Without loss of generality, consider the case where $\mathbf{C}_2$ has at least one non-zero element. Along the lines of the remainder theorem for polynomials in a scalar unknown, the following results provide a method of finding the eigenvalues of the unknown matrix **X**.

Lemma 1. Given that **X** satisfies (2.1) and $\mathbf{G}(\lambda) = \lambda^2\mathbf{C}_2 + \lambda\mathbf{C}_1 + \mathbf{C}_0$, $\mathbf{G}(\lambda)$ has the following representation

$$\mathbf{G}(\lambda) = (\lambda\mathbf{I}-\mathbf{X})(\lambda\mathbf{C}_2 + \mathbf{X}\mathbf{C}_2 + \mathbf{C}_1)\,.$$

Proof.

$$(\lambda\mathbf{I} - \mathbf{X})(\lambda\mathbf{C}_2 + \mathbf{X}\mathbf{C}_2 + \mathbf{C}_1) = \lambda^2\mathbf{C}_2 + \lambda\mathbf{X}\mathbf{C}_2 + \lambda\mathbf{C}_1 - \lambda\mathbf{X}\mathbf{C}_2 - \mathbf{X}^2\mathbf{C}_2 - \mathbf{X}\mathbf{C}_{1.}$$

$$= \lambda^2\mathbf{C}_2 + \lambda\mathbf{C}_1 - \mathbf{X}^2\mathbf{C}_2 - \mathbf{X}\mathbf{C}_{1}.$$

But by hypothesis, $\mathbf{X}^2\mathbf{C}_2 + \mathbf{X}\mathbf{C}_1 + \mathbf{C}_0 = \mathbf{0}$, which implies

$$(\lambda\mathbf{I}-\mathbf{X})(\lambda\mathbf{C}_2 + \mathbf{X}\mathbf{C}_2 + \mathbf{C}_1) = \lambda^2\mathbf{C}_2 + \lambda\mathbf{C}_1 + \mathbf{C}_0\,.$$

■

Theorem 1. The characteristic polynomial of every solution of (2.1) divides the following determinental polynomial

$$\mathrm{Det}\,[\,\mathbf{G}(\lambda)\,] = \mathrm{Det}\,[\,\lambda^2\mathbf{C}_2 + \lambda\mathbf{C}_1 + \mathbf{C}_0\,]\,.$$

Proof. From Lemma (1),

$$\mathbf{G}(\lambda) = (\lambda\mathbf{I} - \mathbf{X})(\lambda\mathbf{C}_2 + \mathbf{X}\mathbf{C}_2 + \mathbf{C}_1).$$

Hence on taking the determinant of $\mathbf{G}(\lambda)$

$$\mathrm{Det}\,[\,\mathbf{G}(\lambda)\,] = \mathrm{Det}\,[\,\lambda\mathbf{I} - \mathbf{X}\,]\;\mathrm{Det}\,[\,\lambda\mathbf{C}_2 + \mathbf{X}\mathbf{C}_2 + \mathbf{C}_1\,]$$

Hence the characteristic polynomial of every solution of (2.1) will divide the determinantal polynomial Det [$\mathbf{G}(\lambda)$] .

■

Theorem 2. The family of all possible solutions of (2.1) decomposes into at most $\binom{2n}{n}$ classes, each of which contains a finite number of equivalence classes of solutions. Furthermore the solutions in each eqivalence class are determined from the solutions of linear homogeneous equations.

Proof. Let $L(\lambda) = \mathrm{Det}\,[\,\mathbf{G}(\lambda)\,]$. Clearly $L(\lambda)$ is a polynomial of degree 2n. Let the set of roots of the determinantal polynomial $L(\lambda)$ be denoted by S.

By theorem (1), the set of eigenvalues of every solution of (2.1) is a subset of S. Hence there are at most $\binom{2n}{n}$ sets of roots, each of which possibly corresponds to the eigenvalues of some solution of (2.1). Thus there at most $\binom{2n}{n}$ classes of solutions which are determined by the set of eigenvalues.

Since every square matrix has Jordan canonical form representation, every solution in each class is of the form

$$\mathbf{X}_i = \mathbf{E}_i\mathbf{D}_i\mathbf{E}_i^{-1} \tag{2.2}$$

where $\mathbf{D}_i$ is in Jordan canonical form and the entries of $\mathbf{D}_i$ are obtained from the set of eigenvalues determining the class. Within each class, the equivalence classes of solutions are determined by the equivalence relation given in (2.2). Since there are only finitely many nxn matrices in Jordan canonical form having a given set of n eigen values as the diagonal elements, there are finitely many $\mathbf{D}_i$'s within each class defining the equivalence classes.

Now consider a member $\mathbf{X}_i$ of the equivalence class defined by $\mathbf{D}_i$. Substituting for $\mathbf{X}_i$ in (2.1)

$$\mathbf{E}_i\mathbf{D}_i^2\mathbf{E}_i^{-1}\mathbf{C}_2 + \mathbf{E}_i\mathbf{D}_i\mathbf{E}_i^{-1}\mathbf{C}_1 + \mathbf{C}_0 = \mathbf{0}.$$

Since $\mathbf{E}_i$ is nonsingular

$$\mathbf{D}_i^2\mathbf{E}_i^{-1}\mathbf{C}_2 + \mathbf{D}_i\mathbf{E}_i^{-1}\mathbf{C}_1 + \mathbf{E}_i^{-1}\mathbf{C}_0 = \mathbf{0}.$$

Since $\mathbf{D}_i$, $\mathbf{C}_2$, $\mathbf{C}_1$, $\mathbf{C}_0$ are known the above is a system of n^2 linear homogeneous equations in n^2 unknowns. Every nonsingular solution of the system determines a member of the equivalence class.

■

To illustrate the utility of the above results, we now consider the solutions of matrix quadratic equations which arise in the equilibrium analysis of QBD processes. We consider the important practical case where $\mathbf{Q}$ is irreducible and positive recurrent.

3. Computation of the Rate Matrix of Quasi Birth and Death Processes

The rate matrix $\mathbf{R}$ is the minimal nonnegative solution of the matrix quadratic equation in (1.3). Since (1.3) has the same form as (2.1), the eigenvalues of $\mathbf{R}$ can be obtained from the roots of the determinental polynomial $L(\lambda) = \mathrm{Det}\,[\,\lambda^2\mathbf{A}_2 + \lambda\mathbf{A}_1 + \mathbf{A}_0\,]$. $L(\lambda)$ is a polynomial of degree 2n in λ . Since the irreducible Markov process $\mathbf{Q}$ is positive recurrent, the spectral radius, $\mathbf{sp}(\mathbf{R})$, of $\mathbf{R}$ is strictly less than one. Hence n of the roots of $L(\lambda)$ lie strictly inside the unit circle in the complex plane. But in order to determine the eigenvalues of $\mathbf{R}$

unambiguously from the roots of $L(\lambda)$, it still remains to localize the other n roots to a distinct region of the complex plane. Theorem (3) provides the desired result. Some useful details are developed first.

Definition: A matrix is said to be of *generator-type* if it has negative diagonal and nonnegative off diagonal elements. Furthermore the sum of elements of each row is negative.

■

Lemma 2. Every matrix of generator type is invertible and the inverse is nonpositive.

Proof. Let $\mathbf{C}$ be a matrix of generator type. $\mathbf{C}$ can be written as

$$\mathbf{C} = \mathbf{D} + \mathbf{E},$$

where $\mathbf{D}$ is a diagonal matrix consisting of the diagonal entries of $\mathbf{C}$ and $\mathbf{E}$ has the same off-diagonal elements as $\mathbf{C}$ with zero diagonal elements. Since $\mathbf{C}$ is of generator type, $\mathbf{D}$ is nonsingular. Hence $\mathbf{C} = \mathbf{D}\,[\,\mathbf{I} + \mathbf{D}^{-1}\mathbf{E}\,]$.

Since $\mathbf{C}$ is of generator type, the maximum row sum of $-\mathbf{D}^{-1}\mathbf{E}$ is strictly less than one. But the maximum row sum is an operator norm of a matrix and the maximum eigenvalue of a matrix is less than or equal to any operator norm. Therefore all the eigenvalues of $-\mathbf{D}^{-1}\mathbf{E}$ lie strictly inside the unit circle in the complex plane. So $[\,\mathbf{I}+\mathbf{D}^{-1}\mathbf{E}\,]$ is nonsingular.
Hence a matrix of generator type is non-singular.
Thus

$$\mathbf{C}^{-1} = [\,\mathbf{I}+\mathbf{D}^{-1}\mathbf{E}\,]^{-1}\,\mathbf{D}^{-1}$$

where

$$[\,\mathbf{I}+\mathbf{D}^{-1}\mathbf{E}\,]^{-1} = \mathbf{I} + (\,-\mathbf{D}^{-1}\mathbf{E}\,) + (\,-\mathbf{D}^{-1}\mathbf{E}\,)^2 + \;\ldots\ldots\;.$$

Since $\mathbf{D}^{-1}$ is nonpositive, $(-\mathbf{D}^{-1}\mathbf{E})$ is nonnegative. Thus from the above expansion $[\,\mathbf{I}+\mathbf{D}^{-1}\mathbf{E}\,]^{-1}$ is nonnegative. which implies $\mathbf{C}^{-1}$ is nonpositive.

■

Lemma 3. The matrix $\mathbf{R}\mathbf{A}_2 + \mathbf{A}_1$ is nonsingular and its inverse is nonpositive.

Proof. Consider the discrete parameter chain obtained by considering the process $\tilde{\mathbf{Q}}$ immediately after its successive transitions. Its transition probability matrix is obtained by using the approach described in [1, p.33]. The state transition matrix $\mathbf{P}^{\circ}$ has the form

$$\mathbf{P}^{\circ} = \begin{bmatrix} \mathbf{B}_0^{\circ} & \mathbf{A}_0^{\circ} & \mathbf{0} & \mathbf{0} & \cdots \\ \mathbf{B}_1' & \mathbf{A}_1' & \mathbf{A}_0' & \mathbf{0} & \cdots \\ \mathbf{B}_2' & \mathbf{A}_2' & \mathbf{A}_1' & \mathbf{A}_0' & \cdots \\ \mathbf{B}_3' & \mathbf{A}_3' & \mathbf{A}_2' & \mathbf{A}_1' & \cdots \\ \vdots & \vdots & \vdots & \vdots & \end{bmatrix}.$$

where

$$\mathbf{B}_0^{\circ} = \Delta_0^{-1}\mathbf{B}_0 + \mathbf{I},$$
$$\mathbf{A}_0^{\circ} = \Delta_0^{-1}\mathbf{A}_0,$$

$$\mathbf{B}_\nu' = \Delta^{-1}\mathbf{B}_\nu, \qquad \text{for } \nu \geqq 1,$$

$$\mathbf{A}_\nu' = \Delta^{-1}\mathbf{A}_\nu + \delta_{1\nu}\,\mathbf{I}, \qquad \text{for } \nu \geqq 0,$$
$$\Delta = -\mathrm{diag}(\mathbf{A}_1'), \qquad \Delta_0 = -\mathrm{diag}(\mathbf{B}_0^{\circ}) \text{ and}$$

$$\mathbf{A}' = \mathbf{A}_0' + \mathbf{A}_1' + \mathbf{A}_2', \qquad \mathbf{A}'(\eta) = \mathbf{A}_0' + \mathbf{A}_1'\eta + \mathbf{A}_2'\eta^2.$$

Let $\mathbf{R}^{\circ}$ be the rate matrix of the underlying discrete parameter Markov chain and $\mathbf{u}^*$ be a positive left eigenvector of $\mathbf{R}^{\circ}$ corresponding to the largest eigenvalue η. Then we have

$$\mathbf{u}^*(\mathbf{A}_1' + \mathbf{R}^{\circ}\mathbf{A}_2') = \mathbf{u}^*(\mathbf{A}_1' + \eta\mathbf{A}_2') = \eta^{-1}\mathbf{u}^*[\mathbf{A}'(\eta) - \mathbf{A}_0']$$

Since $\mathbf{R}^{\circ 2}\mathbf{A}_2' + \mathbf{R}^{\circ}\mathbf{A}_1' + \mathbf{A}_0' = \mathbf{R}^{\circ}$ we have $\mathbf{u}^*[\mathbf{A}'(\eta)] = \mathbf{u}^*\eta$. Hence from the above we have

$$\mathbf{u}^*[\mathbf{A}_1' + \mathbf{R}^{\circ}\mathbf{A}_2'] = \mathbf{u}^* - \eta^{-1}\mathbf{u}^*\mathbf{A}_0' \leqq \mathbf{u}^*.$$

Hence $[\,\mathbf{A}_1' + \mathbf{R}^{\circ}\mathbf{A}_2'\,]$ has spectral radius strictly less than one [10, p.484].

The rate matrices $\mathbf{R}^{\circ}$ and $\mathbf{R}$ satisfy the relation [1, p.34],

$$\mathbf{R} = \Delta\mathbf{R}^{\circ}\Delta^{-1}.$$

Hence using the definition of matrices $\mathbf{A}_i'$, we have

$$\begin{aligned}\mathbf{A}_1' + \mathbf{R}^{\circ}\mathbf{A}_2' &= \Delta^{-1}\mathbf{A}_1 + \mathbf{I} + \Delta^{-1}\mathbf{R}\mathbf{A}_2 \\ &= \Delta^{-1}(\,\mathbf{A}_1 + \mathbf{R}\mathbf{A}_2\,) + \mathbf{I}\,.\end{aligned}$$

Let λ° and λ be the eigenvalues of $(\,\mathbf{A}_1' + \mathbf{R}^{\circ}\mathbf{A}_2'\,)$ and $\Delta^{-1}(\,\mathbf{A}_1 + \mathbf{R}\mathbf{A}_2\,)$ respectively. Clearly $\lambda^{\circ} = 1 + \lambda$. Since the spectral radius of $\mathbf{A}_1' + \mathbf{R}^{\circ}\mathbf{A}_2'$ is strictly less than one, $\Delta^{-1}(\,\mathbf{A}_1 + \mathbf{R}\mathbf{A}_2\,)$ is nonsingular. Consequently $(\,\mathbf{A}_1 + \mathbf{R}\mathbf{A}_2\,)$ is nonsingular.

Since $\mathbf{A}_0$, $\mathbf{A}_1$, $\mathbf{A}_2$ are the sub-matrices of the generator matrix $\mathbf{Q}$

$$(\,\mathbf{A}_0 + \mathbf{A}_1 + \mathbf{A}_2\,)\,\mathbf{e} = \mathbf{0},$$

where $\mathbf{e}$ is a column vector of 1's. From lemma (1), for $\lambda = 1$

$$(\,\mathbf{A}_0 + \mathbf{A}_1 + \mathbf{A}_2\,)\,\mathbf{e} = (\,\mathbf{I} - \mathbf{R}\,)(\,\mathbf{A}_2 + \mathbf{A}_1 + \mathbf{R}\mathbf{A}_2\,)\,\mathbf{e} = \mathbf{0}.$$

But $\mathbf{R}$ has spectral radius strictly less than one. Therefore

$$(\,\mathbf{R}\mathbf{A}_2 + \mathbf{A}_1 + \mathbf{A}_2\,)\,\mathbf{e} = \mathbf{0}.$$

Since $\mathbf{A}_2\mathbf{e} \geqq \mathbf{0}$,

$$(\,\mathbf{R}\mathbf{A}_2 + \mathbf{A}_1)\mathbf{e} \leqq \mathbf{0}.$$

Therefore $(\mathbf{R}\mathbf{A}_2 + \mathbf{A}_1)$ has negative diagonal elements and nonnegative off-diagonal elements and the sum of the elements of all the rows is nonpositive. Furthermore since it is invertible, by using an identical argument as in the above lemma it can be seen that the inverse is nonpositive.

■

Theorem 3. (**Localization Theorem**)

The roots of the polynomial $L(\lambda)$ which are not the eigenvalues of the rate matrix are all on or outside the unit circle in the complex plane.

Proof. Let $\mathbf{M}(\lambda) = \lambda\mathbf{A}_2 + \mathbf{R}\mathbf{A}_2 + \mathbf{A}_1$.
From lemma (1)

$$L(\lambda) = \mathrm{Det}\,[\,\lambda\mathbf{I} - \mathbf{R}\,]\,\mathrm{Det}\,[\,\mathbf{M}(\lambda)\,]\,. \tag{3.2}$$

Hence it is sufficient to show that all the roots of Det $[\,\mathbf{M}(\lambda)\,]$ are all on or outside the unit circle in the complex plane.

Let $\lambda = 0$ be a root of $L(\lambda)$. Then $\lambda = 0$ is not a root of Det($\mathbf{M}(\lambda)$), since by lemma (3), $\mathbf{R}\mathbf{A}_2 + \mathbf{A}_1$ is nonsingular. Now let ν be a nonzero root of Det $[\,\mathbf{M}(\lambda)\,]$. Then

$$\mathbf{M}(\nu)\,\mathbf{f} = \mathbf{0} \qquad \text{for some} \quad \mathbf{f} \neq \mathbf{0}\,.$$

Equivalently

$$\nu \mathbf{A}_2 \mathbf{f} = -(\mathbf{R}\mathbf{A}_2 + \mathbf{A}_1)\mathbf{f}$$

Therefore $\mathbf{E}\,\mathbf{f} = \mu\,\mathbf{f}$,
where $\mu = 1/\nu$ and $\mathbf{E} = -(\mathbf{R}\mathbf{A}_2 + \mathbf{A}_1)^{-1}\mathbf{A}_2$. Hence the inverse of every root of Det $[\,\mathbf{M}(\lambda)\,]$ is an eigenvalue of $\mathbf{E}$.

Since by lemma (3), $(\mathbf{R}\mathbf{A}_2 + \mathbf{A}_1)^{-1}$ is nonpositive, $\mathbf{E}$ is nonnegative. Hence by Frobenius-Perron theorem [8, p.476], $\mathbf{E}$ has an eigenvalue α which is real, positive and simple. For any other eigenvalue λ of $\mathbf{E}$, $|\lambda| \leqq \alpha$.
Since $(\mathbf{A}_0 + \mathbf{A}_1 + \mathbf{A}_2)\mathbf{e} = \mathbf{0}$, by lemma (1), $(\mathbf{R}\mathbf{A}_2 + \mathbf{A}_1 + \mathbf{A}_2)\,\mathbf{e} = \mathbf{0}$. Hence $\mathbf{E}\mathbf{e} = \mathbf{e}$ and $\mathbf{E}$ is a stochastic matrix and $\alpha = 1$. Equivalently, the roots of the polynomial L(λ) which are not the eigenvalues of the rate matrix, **R** are all on or outside the unit circle in the complex plane.
■

As the following Lemma shows, the localization theorem also provides a method of finding the left eigenvectors of **R**.

Lemma 4. The row vector $\bar{\mathbf{u}}$ is a left eigenvector corresponding to an eigenvalue, λ of **R** if and only if

$$\bar{\mathbf{u}}\,(\lambda^2\mathbf{A}_2 + \lambda\mathbf{A}_1 + \mathbf{A}_0) = \mathbf{0}. \tag{3.3}$$

Proof. First consider a zero eigenvalue of **R** and let $\bar{\mathbf{u}}$ be the corresponding left eigenvector. Since **R** satisfies (1.3),

$$\bar{\mathbf{u}}(\mathbf{R}^2\mathbf{A}_2 + \mathbf{R}\mathbf{A}_1 + \mathbf{A}_0) = \mathbf{0},$$

which implies $\bar{\mathbf{u}}\mathbf{A}_0 = \mathbf{0}$. Thus every vector in the left null space of **R** is also in the left null space of $\mathbf{A}_0$. From (1.3),

$$\mathbf{R}(\,\mathbf{R}\mathbf{A}_2 + \mathbf{A}_1\,) = -\mathbf{A}_0.$$

Since $\mathbf{R}\mathbf{A}_2 + \mathbf{A}_1$ is nonsingular by lemma (3) and since the rank of a matrix is not changed on multiplication by a nonsingular matrix, we have

$$\mathrm{Rank}(\mathbf{R}) = \mathrm{Rank}(\mathbf{A}_0).$$

Hence the left null spaces of **R** and $\mathbf{A}_0$ coincide. Equivalently $\bar{\mathbf{u}}$ is a left eigenvector corresponding to a zero eigenvalue of **R** if and only if

$$\bar{\mathbf{u}}\mathbf{A}_0 = \mathbf{0}.$$

Now consider a nonzero eigenvalue λ of **R**. Since **R** satisfies (1.3) and (1.3) has the same form as (2.1), Lemma (1) can be applied. Hence

$$\lambda^2\mathbf{A}_2 + \lambda\mathbf{A}_1 + \mathbf{A}_0 = (\,\lambda\mathbf{I} - \mathbf{R}\,)\,\mathbf{M}(\lambda)\,. \tag{3.4}$$

Necessity follows from the definition of left eigenvector and from (1.3). Now suppose that $\bar{\mathbf{u}}$ satisfies (3.3). On using (3.4)

$$\bar{\mathbf{u}}\,(\,\lambda\mathbf{I} - \mathbf{R}\,)\,\mathbf{M}(\lambda) = \mathbf{0}.$$

R has spectral radius strictly less than one and so the absolute value of λ is strictly less than one. By the localization theorem, $\mathbf{M}(\lambda)$ is nonsingular if the absolute value of λ is strictly less than one. Therefore we have

$$\bar{\mathbf{u}}\,(\,\lambda\mathbf{I} - \mathbf{R}\,) = \mathbf{0}.$$

Hence $\bar{\mathbf{u}}$ is a left eigenvector of **R**.
■

In the remainder of this paper, it is assumed that the rate matrix, **R** is diagonalizable for the sake of simplicity and clarity. Let **S** be an n×n matrix whose rows are the left eigenvectors of **R** and **T** be the corresponding matrix whose columns are the normalized right eigenvectors.

The following theorems provide two methods of finding the right eigenvectors and thus the rate matrix.

Theorem 4. The spectral representation of the rate matrix **R**, the minimal nonnegative solution of the matrix quadratic equation (1.3) can be found.

Proof. The rate matrix **R** has the spectral representation of the form

$$\mathbf{R} = \mathbf{TDT}^{-1}.$$

where **D** is a diagonal matrix with the eigenvalues of **R** on the diagonal and the columns of the matrix are the right eigenvectors of **R**.

Using the localization theorem the eigenvalues of **R** can be found from the roots of the determinental polynomial $L(\lambda)$. As shown in Lemma 4, a left eigenvector corresponding to a nonzero eigenvalue can be determined by finding a vector in the left null space of the matrix $(\lambda^2\mathbf{A}_2 + \lambda\mathbf{A}_1 + \mathbf{A}_0)$. Since **R** is diagonalizable, the set of n left eigenvectors are linearly independent and hence the matrix **S** of left eigenvectors is nonsingular. Since

$$\mathbf{S} = \mathbf{T}^{-1}$$

the matrix **T** can be found by inverting **S** and thus a spectral representation of the rate matrix can be found.

■

The following theorem shows that knowing the eigenvalues and the left eigenvectors, the problem of finding **R**, the minimal nonnegative solution of matrix quadratic equation, can be formulated as a linear programming problem. This method which is a variation of the method used in [5], should have desirable numerical properties.

Theorem 5. Suppose that the eigenvalues and the left eigenvectors of **R** are known and that **R** is diagonalizable. Let λ_i be the i'th nonzero eigenvalue of **R** and $\bar{\mathbf{u}}_i$ be the corresponding left eigenvector. Let $\mathbf{x}_i$, $1 \leqq i \leqq p$ be unknown column vectors of dimension n. Then the right eigenvectors of the rate matrix are the solution to the following linear program.

$$\min_{\mathbf{x}_k,\ k=1,2,\ldots,p} \quad \sum_{k=1}^{p} \bar{\mathbf{e}} \cdot \left(\lambda_k \mathbf{x}_k \bar{\mathbf{u}}_k\right) \mathbf{e},$$

where **e** is a column of n 1's with transpose $\bar{\mathbf{e}}$, subject to the linear constraints:

$$\sum_{j=1}^{p} \lambda_j^2 \mathbf{x}_j \bar{\mathbf{u}}_j \mathbf{A}_2 + \sum_{j=1}^{p} \lambda_j \mathbf{x}_j \bar{\mathbf{u}}_j \mathbf{A}_1 + \sum_{j=1}^{p} \mathbf{x}_j \bar{\mathbf{u}}_j \mathbf{A}_0 = \mathbf{0}. \tag{3.5}$$

and

$$\sum_{j=1}^{p} \lambda_j \mathbf{x}_j \bar{\mathbf{u}}_j \geqq \mathbf{0}\ ; \qquad \bar{\mathbf{u}}_i \mathbf{x}_j = \delta_{ij} \tag{3.6}$$

Furthermore once the optimal vectors $\mathbf{x}_i{}^{\circ}$ are found

$$\mathbf{R} = \sum_{i=1}^{p} \lambda_i \mathbf{x}_i{}^{\circ} \bar{\mathbf{u}}_i\ .$$

Proof. See [1].

To illustrate the utility of the above results, a specific QBD process which models a slotted ALOHA network is analyzed.

4. QBD Model of Slotted ALOHA Networks

A. Assumptions and Model

The channel activity on the slotted ALOHA channel is modeled using a continuous time vector Markov process $z_t = (n_t, cs_t)$, where n_t and cs_t indicate the number of busy users and channel state respectively at time t. In the case considered here, there are four possible channel states, namely the idle state

$(cs_t = 0)$,

the single user transmission state $(cs_t = 1)$, the collision state $(cs_t = c)$ and the guard state $(cs_t = g)$.

The slot length and synchronization time (guard state) random variables have probability distributions of phase type. The PH distributions are chosen to approximate the actual distribution. For example the length of a slot has a k-stage Erlangian distribution with large k so that a constant slot length is approximated. For simplicity we consider k=1. But the analysis can easily be extended to the case where k is arbitrary [1].

Each user receives packets at rate λ, with exponential interarrival times of mean length $1/\lambda$ and the users are mutually independent. When there are n busy users on the channel, each user transmits a packet in the slot following the next guard slot with probability $p_{n,r}$. We consider the infinite population case and constant retransmission loading policy in which case $np_{n,r} = \gamma$.

The timing and state transition diagrams are shown in Figs. 1 and 2, respectively. The global balance equations can be written down by inspection from the state transition diagram in Fig. 2. Using the global balance equations, the generator of the Markov process can be shown to be of the form in (1.1), where the $\mathbf{A}_i$'s are square matrices of order 4 and are given by

$$A_0 = \begin{bmatrix} G & 0 & 0 & 0 \\ 0 & G & 0 & 0 \\ 0 & 0 & G & 0 \\ 0 & 0 & 0 & G \end{bmatrix} \quad A_1 = \begin{bmatrix} -\lambda_s - G & 0 & 0 & \lambda_s \\ 0 & -\lambda_s - G & 0 & 0 \\ 0 & 0 & -\lambda_s - G & \lambda_s \\ p_1\lambda_g & p_2\lambda_g & p_3\lambda_g & -\lambda_g - G \end{bmatrix} \quad A_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda_s \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

where,

$G = \lim_{\substack{N \to \infty \\ \lambda \to 0}} (N - n)\,\lambda$, and $1/\lambda_s$ and $1/\lambda_g$ are the mean values of slot length and synchronization time (guard state) random variables and p_1, p_2, p_3 are the probabilities that 0, 1 or 2 users transmit in a given slot.

B. Peformance Evaluation

Stability: The slotted ALOHA network is defined to be stable when the number of busy users remains finite for all time or equivalently the QBD process is recurrent nonnull. It is shown in [3], that a necessary and sufficient condition for stability is that $\mathrm{sp}(\mathbf{R})<1$. An equivalent condition is given by the follwing theorem.

Theorem 6. The spectral radius of $\mathbf{R}$ is less than unity and the QBD process is recurrent nonnull if and only if

$$\pi \mathbf{A}_0 \mathbf{e} < \pi \mathbf{A}_2 \mathbf{e}$$

where π is uniquely determined by $\pi \mathbf{A} = \mathbf{0}$ and $\pi \mathbf{e} = 1$, with $\mathbf{A} = \mathbf{A}_0 + \mathbf{A}_1 + \mathbf{A}_2$.
Proof. see [3].
Thus the stability of the slotted ALOHA network can be determined by verifying the above condition.

Throughput-Delay Characteristics: Once the rate matrix of the underlying QBD process is computed, the distribution of the number of busy users can easily be found. The probability that n users are busy is

$$\overline{\pi}_n \mathbf{e} = \overline{\pi}_2 \mathbf{R}^{n-2} \mathbf{e} \quad n \geqq 2.$$

The probability that there are no busy users or equivalently that the network is idle is $\overline{\pi}_0\mathbf{e}$ and the probability that there is a single busy user is $\overline{\pi}_1\mathbf{e}$. The expected number of busy users E(N) is

$$E(N) = \bar{\pi}_1\mathbf{e} + \bar{\pi}_2\left[\sum_{i=2}^{\infty} i\mathbf{R}^{i-2}\right]\mathbf{e}.$$

In a stable network sp($\mathbf{R}$)<1. Hence all the eigenvalues of **R** are within the unit circle. Therefore

$$E(N) = \bar{\pi}_1\mathbf{e} + \bar{\pi}_2(2\mathbf{I} - \mathbf{R})(\mathbf{I} - \mathbf{R})^{-2}\mathbf{e}.$$

Now using Little's theorem, an exact closed form expression for the expected delay E(D) is found to be

$$E(D) = \frac{E(N)}{G\lambda_s}$$

where $G\lambda_s$ is the throughput since the network is stable.

5. Conclusions

In this paper, the solutions of a matrix quadratic equation are explored. The techniques developed have demonstrated that it is possible to find the spectral representation of the rate matrix, the minimal nonnegative solution of a matrix quadratic equation. The results are extended to matrix polynomial equations in [2], which arise in the equilibrium analysis of G/M/1 type Markov processes used in stochastic models appearing in communication, queuing and inventory theories. The practical utility of the results is illustrated through the analysis of a QBD model of slotted ALOHA networks. The analytical framework can be extended to a model of spread spectrum slotted ALOHA packet radio networks.

References

[1] G. Rama Murthy and E. J. Coyle, "Matrix quadratic equations and quasi-birth-and-death models of multiple access networks, " Manuscript under preparation.

[2] G. Rama Murthy and E.J. Coyle, "Nonlinear matrix equations and matrix geometric solutions in stochastic models," Manuscript under preparation.

[3] M. F. Neuts, Matrix-Geometric Solutions in Stochastic Models, Johns Hopkins Univ. Press, Baltimore, 1981.

[4] R. V. Evans, "Geometric distribution in some two-dimensional queuing systems," *Operations Research,* vol 15, pp. 830-46, 1967.

[5] S. L. Beuerman and E. J. Coyle, "State space expansions and the limiting behaviour of QBD processes," Advances in Applied Probability, to appear in June 1989.

[6] E. J. Coyle and B. Liu, "A matrix representation of CSMA/CD networks," *IEEE Transactions on Communications,* vol. COM-33, pp. 53-64, Jan 1985.

[7] E. J. Coyle and B. Liu, "Finite population CSMA/CD Networks," *IEEE Transactions on Communications,* vol. COM-31, pp. 1247-1251, Nov 1983.

[8] J. M. Ferrandiz and A. A. Lazar, "Geometric analysis of quasi-birth -and-death processes by flow reduction," *Conference on Information Sciences and Systems,* Johns Hopkins University, Baltimore, MD, March 1988.

[9] V. Ramaswami and G. Latouche, "A general class of Markov Processes with explicit matrix geometric solutions, OR Spektrum, vol.8, pp. 209-218, 1986.

[10] S. Karlin and H. M. Taylor, A first course in stochastic processes, Academic Press, New York, 1975.

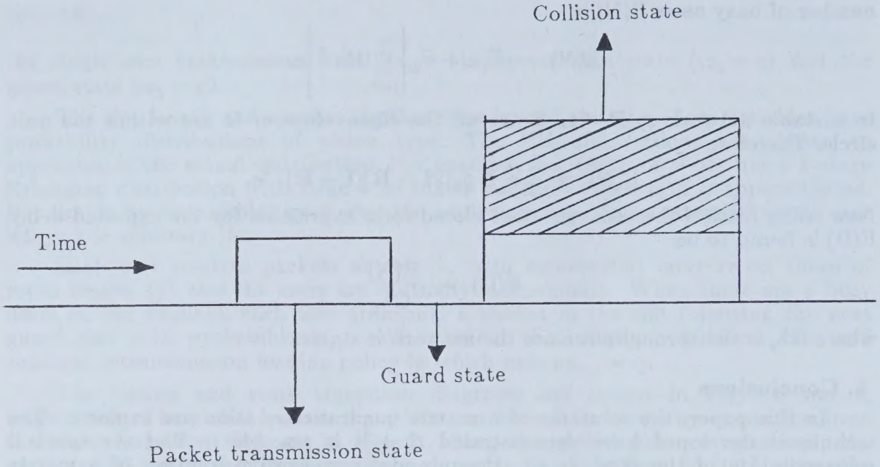

Collision state
Time
Guard state
Packet transmission state

Fig. 1 Typical channel activity of narrow band slotted ALOHA packet network.

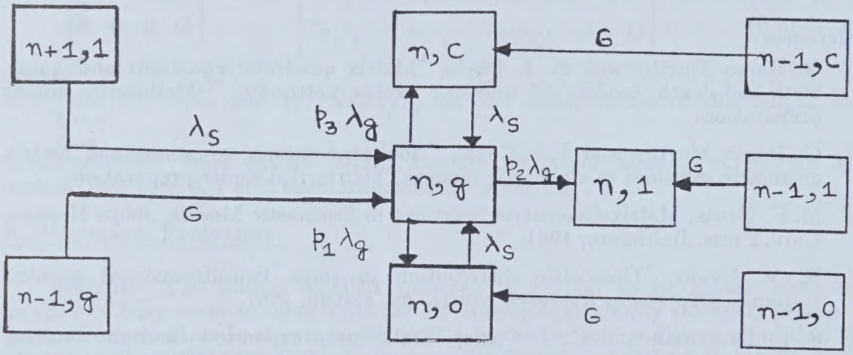

n+1,1
n, c
n-1, c
G
$p_3 \lambda_g$
λ_s
λ_s
$p_2 \lambda_g$
G
n, g
n, 1
n-1, 1
G
$p_1 \lambda_g$
λ_s
n-1, g
n, 0
G
n-1, 0

Fig. 2 State transition diagram for slotted ALOHA protocol. All possible transitions into and between all states in which n users are busy is shown along with the rate at which each transition takes place.

ON FUSION IN ESTIMATION THEORY

ERIC B. HALL
Department of Electrical and Computer Engineering
The University of Texas at Austin
Austin, Texas 78712

ALAN E. WESSEL
Department of Mathematics
Santa Clara University
Santa Clara, California 95053

GARY L. WISE
Department of Electrical and Computer Engineering
and Department of Mathematics
The University of Texas at Austin
Austin, Texas 78712

ABSTRACT

This paper considers methods of fusing or combining various estimates of a fixed random variable so as to obtain a single good estimate of the fixed random variable. These estimates include conditional expectation as well as projection onto closed linear manifolds spanned by the data. We present several examples, using common distributions, in which virtually any method of fusion would be useless in approximating the random variable of interest. These examples expose several fallacies in the literature. Finally, we present a constructive approach to this problem based on more practical concerns.

INTRODUCTION

In this paper we consider methods of fusing or combining a finite number of estimates of a fixed second order random variable X in order to achieve a single "best" estimate of X. Our concern throughout this paper is directed toward minimizing the mean square error. In this regard, we recall [1] that it is necessary to take versions of conditional expectations which are Borel measurable functions of the conditioned random variables, and we do so throughout the paper.

As an example of fusion in estimation, if Y_1 and Y_2 are random variables, how might $E[X|Y_1]$ and $E[X|Y_2]$ be fused so as to obtain a good approximation to $E[X|Y_1, Y_2]$? Notice that the statistical knowledge required to calculate $E[X|Y_1]$ and $E[X|Y_2]$ is less than that required to calculate $E[X|Y_1, Y_2]$, since $E[X|Y_1]$ and $E[X|Y_2]$ can be obtained from the appropriate bivariate distributions, whereas $E[X|Y_1, Y_2]$ in general cannot. Also, Y_1 and Y_2 might not be simultaneously available to the person desiring $E[X|Y_1, Y_2]$, a situation occurring in the usual context of distributed estimation in which a central location desires to construct a good estimate based on the estimates obtained by a set of spatially distinct field observers. Although the problem of fusion in estimation theory has been addressed by other authors (see, for example, [2-5], each of which neglected the caveat mentioned in [1]), most important questions in this area remain heretofore unanswered. In this paper we put the problem of fusion in estimation on a rigorous basis and expose several problems associated with an overly cavalier approach to this subject.

DEVELOPMENT

Let X be a second order random variable and n be a positive integer. The following lemma and associated corollary will prove useful in this work. In particular, these results justify the estimation of $E[X|Y_1, Y_2, \ldots, Y_n]$ instead of X.

Lemma 1: Let m be a positive integer. Consider a probability space $(\Omega, \mathcal{F}, P)$, a random variable X belonging to $L_2(\Omega, \mathcal{F}, P)$, and a random vector Y taking values in $\mathbf{R}^m$ and consisting of random variables defined on $(\Omega, \mathcal{F}, P)$. Consider a nonempty subset $\mathcal{G}$ of the set of all Borel measurable functions mapping $\mathbf{R}^m$ to $\mathbf{R}$, and let $\{g_1, g_2, \ldots\}$ be a sequence of elements from $\mathcal{G}$. Then $\lim_{j\to\infty} E[(X-g_j(Y))^2] = \inf_{g\in\mathcal{G}} E[(X-g(Y))^2]$ if and only if $\lim_{j\to\infty} E[(E[X|Y]-g_j(Y))^2] = \inf_{g\in\mathcal{G}} E[(E[X|Y]-g(Y))^2]$.

Proof: Let $g:\mathbf{R}^m\to\mathbf{R}$ be a Borel measurable function and note that $E[(X-g(Y))^2]$ $= E[(X - E[X|Y] + E[X|Y] - g(Y))^2] = E[(X-E[X|Y])^2] + E[(E[X|Y]-g(Y))^2]$ $+ 2\,E[\,(E[X|Y]-g(Y))\,E[(X-E[X|Y])|Y]\,] = E[(X-E[X|Y])^2] + E[(E[X|Y]-g(Y))^2]$. Thus, $\lim_{j\to\infty} E[(X-g_j(Y))^2] = E[(X-E[X|Y])^2] + \lim_{j\to\infty} E[(E[X|Y]-g_j(Y))^2]$, and, $\inf_{g\in\mathcal{G}} E[(X-g(Y))^2] = E[(X-E[X|Y])^2] + \inf_{g\in\mathcal{G}} E[(E[X|Y]-g(Y))^2]$. Therefore, it follows immediately that $\lim_{j\to\infty} E[(X-g_j(Y))^2] = \inf_{g\in\mathcal{G}} E[(X-g(Y))^2]$ if and only if $\lim_{j\to\infty} E[(E[X|Y]-g_j(Y))^2] = \inf_{g\in\mathcal{G}} E[(E[X|Y]-g(Y))^2]$. Q.E.D.

The following corollary now follows easily from Lemma 1.

Corollary 2: Let m be a positive integer. Consider a probability space $(\Omega, \mathcal{F}, P)$, a random variable X belonging to $L_2(\Omega, \mathcal{F}, P)$, and a random vector Y taking values in $\mathbf{R}^m$ and consisting of random variables defined on $(\Omega, \mathcal{F}, P)$. Consider a nonempty subset $\mathcal{G}$ of the set of all Borel measurable functions mapping $\mathbf{R}^m$ to $\mathbf{R}$ and an element h in $\mathcal{G}$. Then

$$E[(E[X|Y]-h(Y))^2] = \inf_{g\in \mathcal{G}} E[(E[X|Y]-g(Y))^2] \text{ if and only if}$$
$$E[(X-h(Y))^2] = \inf_{g\in \mathcal{G}} E[(X-g(Y))^2].$$

Given a family $\mathcal{C}$ of random variables in $L_2(\Omega, \mathcal{F}, P)$, we will let $\mathcal{L}(\mathcal{C})$ denote the subspace of $L_2(\Omega, \sigma(\mathcal{C}), P)$ given by the $L_2(\Omega, \mathcal{F}, P)$ closure of the set of all finite linear combinations of elements of $\mathcal{C}$. Further, we will let $\hat{E}[X|\mathcal{C}]$ denote the orthogonal projection of $X \in L_2(\Omega, \mathcal{F}, P)$ onto the subspace $\mathcal{L}(\mathcal{C})$.

We first consider the fusion of linear estimates of X. For a positive integer $n \geq 2$, consider a family of random variables $\{Y_1, \ldots, Y_n\}$ from $L_2(\Omega, \mathcal{F}, P)$, and let $Z_i \equiv \hat{E}[X|Y_i]$. In other words, Z_i is the orthogonal projection of X onto $\mathcal{L}(Y_i)$. Assume that we wish to fuse the random variables $\{Z_1, \ldots, Z_n\}$ in such a way so as to estimate X based on a linear combination of elements from this family. The following theorem analyzes a special case of this situation.

Theorem 3: Let n be a positive integer greater than one, and consider a probability space $(\Omega, \mathcal{F}, P)$, a random variable $X \in L_2(\Omega, \mathcal{F}, P)$, and a family $\{Y_1, \ldots, Y_n\}$ of random variables from $L_2(\Omega, \mathcal{F}, P)$. Further, let $Z_i \equiv \hat{E}[X|Y_i]$ and assume that the Y_i's are orthogonal, i.e. $E[Y_iY_j] = 0$ for $1 \leq i, j \leq n$ and $i \neq j$, and such that $E[Y_i^2] > 0$ for each i. Then the best L_2 estimate of X based on linear operations on elements from the family $\{Z_1, \ldots, Z_n\}$ is $Z_1 + \ldots + Z_n$.

Proof: By definition, we know that $\hat{E}[X|Y_1, \ldots, Y_n]$, the best L_2 estimate of X based on linear operations on elements from the family $\{Y_1, \ldots, Y_n\}$, is given by $a_1Y_1 + \ldots + a_nY_n$ for some choice of the a_i's. Since $(X - \hat{E}[X|Y_1, \ldots, Y_n])$ is orthogonal to Y_i for each i, and since $E[Y_iY_j] = 0$ for $1 \leq i, j \leq n$ and $i \neq j$, we see that $a_i = E[XY_i]/E[Y_i^2]$ for each positive integer $i \leq n$. Finally, since $(X-\hat{E}[X|Y_i])$ is orthogonal to Y_i for each positive integer $i \leq n$, we see that $\hat{E}[X|Y_i] = E[XY_i]\, Y_i / E[Y_i^2]$.
Q.E.D.

Note that in the case where $X, Y_1, Y_2, \ldots, Y_n$ form a Gaussian random process, then an orthogonal projection of X onto $L_2(\Omega, \sigma(Y_i), P)$ is actually an orthogonal projection of X onto the affine span of Y_i, and an orthogonal projection of X onto $L_2(\Omega, \sigma(Y_1, Y_2, \ldots, Y_n), P)$ is actually an orthogonal projection of X onto the affine span of $Y_1, Y_2, \ldots, Y_n$. Along this line of reasoning, we have the following.

Example 1: Let the sequence $\{X, Y_1, \ldots, Y_n\}$ denote a zero mean Gaussian process with positive variances and assume that the observations $Y_1, \ldots, Y_n$ are mutually independent. Let $d_i \equiv E[XY_i]/E[Y_i^2]$ and note that $W_i \equiv E[X|\sigma(Y_i)] = d_iY_i$ a.s. Also, note that $E[X|\sigma(Y_1, \ldots, Y_n)] = d_1Y_1 + \ldots + d_nY_n$ a.s. since the Y_i's are mutually independent. Thus, for this special case, the best L_2 estimate of $E[X|\sigma(Y_1, \ldots, Y_n)]$ based on linear operations on the W_i's is $\sum_{i=1}^{n} W_i$, which in this case equals $E[X|\sigma(Y_1, \ldots, Y_n)]$ a.s.

With Example 1 as motivation, we present the following.

Theorem 4: Let n be an integer greater than one. Consider a probability space $(\Omega, \mathcal{F}, P)$, a random variable $X \in L_2(\Omega, \mathcal{F}, P)$, and a family $\{Y_1, \ldots, Y_n\}$ of random variables defined on $(\Omega, \mathcal{F}, P)$. Further, for positive integers i not greater than n, let $W_i \equiv E[X|Y_i]$. Then the best L_2 estimate of X based on linear operations on elements from the family $\{W_1, \ldots, W_n\}$ is $\alpha_1W_1 + \ldots + \alpha_nW_n$ where the α_i's are given by a solution, which always exists, to the following system of equations:

$$\begin{bmatrix} E[W_1^2] & E[W_2W_1] & \cdots & E[W_nW_1] \\ E[W_1W_2] & E[W_2^2] & \cdots & E[W_nW_2] \\ \vdots & \vdots & \ddots & \vdots \\ E[W_1W_n] & E[W_2W_n] & \cdots & E[W_n^2] \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} E[W_1^2] \\ E[W_2^2] \\ \vdots \\ E[W_n^2] \end{bmatrix}.$$

Proof: First, define $W \equiv E[X|\sigma(Y_1, \ldots, Y_n)]$, and note that W represents the best L_2 estimate of X which can made based upon a Borel measurable transformation of the Y_i's. In the context of Corollary 2, let $\mathcal{G}$ equal the set of all real valued Borel measurable functions mapping $\mathbf{R}^n$ to $\mathbf{R}$ of the form $g(y_1, \ldots, y_n) = \beta_1g_1(y_1) + \ldots + \beta_ng_n(y_n)$ where $g_i:\mathbf{R}\rightarrow\mathbf{R}$ is the a.e. uniquely defined Borel measurable function such that $g_i(Y_i) = E[X|\sigma(Y_i)]$ a.s. and $\beta_i \in \mathbf{R}$ for each positive integer $i \leq n$. Corollary 2 then implies that finding an element in $\mathcal{G}$ which minimizes $E[(X-g(Y_1, \ldots, Y_n))^2]$ for $g \in \mathcal{G}$ is equivalent to finding an element in $\mathcal{G}$ which minimizes $E[(W-g(Y_1, \ldots, Y_n))^2]$ for $g \in \mathcal{G}$. Recall that $\hat{E}[W|W_1, \ldots, W_n]$ is the a.e. unique element from $\mathcal{L}(W_1, \ldots, W_n)$ which minimizes $E[(W-Z)^2]$ over all $Z \in \mathcal{L}(W_1, \ldots, W_n)$. Thus, from the definition of $\mathcal{G}$, we see that $\hat{E}[W|W_1, \ldots, W_n] = \alpha_1W_1 + \ldots + \alpha_nW_n$ determines an a.e. unique element of $\mathcal{G}$ which minimizes $E[(W-g(Y_1, \ldots, Y_n))^2]$ over all $g \in \mathcal{G}$.

Since $(W-\hat{E}[W|W_1, \ldots, W_n])$ is orthogonal to W_i for each positive integer $i \leq n$ we see that the α_i's may be found by solving the following system of equations:

$$\begin{bmatrix} E[W_1^2] & E[W_2W_1] & \cdots & E[W_nW_1] \\ E[W_1W_2] & E[W_2^2] & \cdots & E[W_nW_2] \\ \vdots & \vdots & \ddots & \vdots \\ E[W_1W_n] & E[W_2W_n] & \cdots & E[W_n^2] \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} E[WW_1] \\ E[WW_2] \\ \vdots \\ E[WW_n] \end{bmatrix}$$

Notice that the corollary to Theorem 4.11 in [7, p.80] implies that a solution to this system exists. The conclusion of Theorem 4 now follows by noticing that $E[WW_i] =$

$E[\, E[X|\sigma(Y_1, \ldots, Y_n)]\, E[X|\sigma(Y_i)]\,] = E[\, E[\, E[X|\sigma(Y_1, \ldots, Y_n)]\, E[X|\sigma(Y_i)] \mid \sigma(Y_i)\,]\,]$

$= E[\, E[X|\sigma(Y_i)]\, E[\, E[X|\sigma(Y_1, \ldots, Y_n)] \mid \sigma(Y_i)\,]\,] = E[\, E[X|\sigma(Y_i)]\, E[X|\sigma(Y_i)]\,]$

$= E[W_i^2]$. Q.E.D.

Even though a solution to the system of equations considered in Theorem 4 always exists, as noted in Examples 2 - 5, given below, the best linear combination of the conditional expectations may fail to be a good estimate.

Consider the following situation. Let X be a second order random variable, let n be a positive integer, and let $Y_1, Y_2, \ldots, Y_n$ be n random variables. How might $E[X|Y_1]$, $E[X|Y_2], \ldots,$ and $E[X|Y_n]$ be fused so as to approximate $E[X|Y_1, Y_2, \ldots, Y_n]$? In particular, under what conditions does $E[X|\, E[X|Y_1], E[X|Y_2], \ldots, E[X|Y_n]]$, the best L_2 estimate of X as a Borel measurable function of $E[X|Y_1], E[X|Y_2], \ldots,$ and $E[X|Y_n]$, approximate $E[X|Y_1, Y_2, \ldots, Y_n]$ well? As the following examples indicate, there are numerous subtleties which arise in this context and which have apparently been neglected by many researchers.

We note that even if $E[X|Y_1] = E[X|Y_2] = \ldots = E[X|Y_n]$ a.s., $E[X|Y_1, Y_2, \ldots, Y_n]$ could be wildly different from $E[X|Y_1]$. Further, it might so happen that there does not exist a linear combination of $E[X|Y_1], E[X|Y_2], \ldots,$ and $E[X|Y_n]$ which is a very good approximation to $E[X|Y_1, Y_2, \ldots, Y_n]$.

Example 2: Let $\Omega = [0, 1]$, $\mathcal{F}$ denote the Borel subsets of Ω, and P denote Lebesgue measure on $\mathcal{F}$. Let A be a positive real number, n = 2, $\sigma(Y_1) = \sigma([0, 1/2))$, $\sigma(Y_2) = \sigma([1/4, 3/4))$, and $X(\omega) = A$ for $\omega \in [0, 1/4) \cup [1/2, 3/4)$ and $X(\omega) = -A$ for $\omega \in [1/4, 1/2) \cup [3/4, 1]$. Then it straightforwardly follows that $E[X|Y_1] = E[X|Y_2] = 0$ a.s., but $E[X|Y_1, Y_2] = X$ a.s. Notice that in this special case, any linear combination of $E[X|Y_1]$ and $E[X|Y_2]$ yields an estimate equal to 0 a.s., resulting in a mean square error in approximating X of A^2, which can exceed any preassigned real number. Recalling that $E[X|Y_1]$ and $E[X|Y_2]$, respectively, are $\sigma(Y_1)$-measurable and $\sigma(Y_2)$-measurable, we see that $E[X|Y_1] = E[X|Y_2] = 0$ pointwise in ω; similarly, we see that $E[X|Y_1, Y_2] = X$

pointwise in ω. Thus we see that it is fruitless to attempt to approximate X based on any function of $E[X|Y_1]$ and $E[X|Y_2]$.

Note that in Example 2, Y_1 and Y_2 are independent, X and Y_1 are independent, and X and Y_2 are independent. This might have lead an unwary, overly cavalier investigator to perhaps assert that $E[X|Y_1, Y_2] = E[X]$ a.s., or perhaps that $E[X|Y_1, Y_2] = E[X|Y_i]$ a.s. for $i = 1$ or 2. Each of these assertions, which happen to be equivalent in the setting of Example 2, is incorrect.

Note that Example 2 concerned simple random variables. However, the phenomenon exhibited in Example 2 can hold for non-simple random variables, as shown in the following three examples, which involve more commonplace distributions of random variables.

Example 3: Let Y_1 and Y_2 be independent Gaussian random variables defined on the same probability space, each having zero mean and unit variance. Let $X = Y_1Y_2$. Then it straightforwardly follows that $E[X|Y_1] = E[X|Y_2] = 0$ a.s., but $E[X|Y_1, Y_2] = X$ a.s.

Example 4: Let Y_1 and Y_2 be independent random variables defined on the same probability space, such that Y_1 is uniformly distributed on $[-1, 1]$ and Y_2 has a probability density function given by $f(x) = \frac{x^2}{\sqrt{2\pi}} \exp(\frac{-x^2}{2})$, and let $X = Y_1Y_2$. It can be shown that X has a Gaussian distribution with zero mean and unit variance. It then again straightforwardly follows that $E[X|Y_1] = E[X|Y_2] = 0$ a.s., but $E[X|Y_1, Y_2] = X$ a.s. We note that this example provides a counterexample to a conclusion stated in [3].

Example 5: We note [6] that for any positive integer $n > 1$, there exists an underlying probability space on which $n + 1$ random variables $X, Y_1, \ldots, Y_n$ are defined such that the set $\{X, Y_1, \ldots, Y_n\}$ is not mutually Gaussian and not mutually independent, yet any proper subset of $\{X, Y_1, \ldots, Y_n\}$ is mutually independent, mutually Gaussian, and identically distributed with zero mean and unit variance. For any proper subset D of $\{Y_1, \ldots, Y_n\}$, we note that $E[X|D] = 0$ a.s. since X is independent of D. However, for the example exhibited in [6], it can be straightforwardly shown that $E[X| Y_1, \ldots, Y_n] = \frac{1}{2\sqrt{2}} Y_1 \cdots Y_n \exp[-((Y_1)^2 + (Y_2)^2 + \ldots + (Y_n)^2)/2]$ a. s. Thus, it would not be reasonable to attempt to estimate $E[X| Y_1, \ldots, Y_n]$ based upon $E[X|D]$ where D ranges over all proper subsets of $\{Y_1, \ldots, Y_n\}$. Note that even the ubiquitous Gaussian assumption does not come to the rescue in this case.

As the above examples suggest, when no restrictions are placed on the data, the problem of fusion in estimation appears hopeless. Note that if $E[X|Y_i] = 0$ a.s. for $i = 1, \ldots, n$, then there exist events $A_1, \ldots, A_n$ of probability one such that $E[X|Y_i] = 0$

pointwise on A_i. Then, since $A = \bigcap_{i=1}^{n} A_i$ is an event with probability one, we see that on A, $E[X|Y_i] = 0$ for $i = 1, \ldots, n$. Thus, any Borel measurable function $f:\mathbf{R}^n \to \mathbf{R}$ is such that $f(E[X|Y_1], \ldots, E[X|Y_n])$ is constant on A, and hence is constant a.s. However, as will be shown next, with an appropriate restriction on the data, almost surely exact fusion is possible.

A CONSTRUCTIVE APPROACH

Let $(\Omega, \mathcal{F}, P)$ be a fixed probability space on which all following random variables will be defined, and let n be a positive integer. We will now consider an approach motivated by more practical concerns. The following notation will be used throughout the remainder of this section. Let $Y_i = X + N_i$ and $S = Y_1 + Y_2 + \ldots + Y_n$, where $X, N_1, N_2, \ldots, N_n$ are mutually independent, N_i represents additive noise, and X is a second order random variable representing the signal of interest. We will let $\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\,dt$. As before, we consider the problem of estimating X via some combination of the $E[X|Y_i]$'s.

Assume that $N_1, N_2, \ldots, N_n$ each possesses a probability density function. Let F_X denote the distribution function of X, and f_{N_i} denote a Borel measurable density function of N_i. Further, notice that via a straightforward change of variable and using the independence assumption, we have that $E[X \mid X+N_1, \ldots, X+N_n] =$

$$\frac{\int_{\mathbf{R}} x \prod_{i=1}^{n} f_{N_i}(X+N_i-x)\, dF_X(x)}{\int_{\mathbf{R}} \prod_{i=1}^{n} f_{N_i}(X+N_i-x)\, dF_X(x)} \text{ a.s.}$$

Now, in addition to the above assumptions, let the N_i's have zero mean, unit variance Gaussian distributions. Notice that in this case $\prod_{i=1}^{n} f_{N_i}(X+N_i-x) =$ $\left(\frac{1}{\sqrt{2\pi}}\right)^n \exp\left(\frac{-1}{2}\left\{nx^2 - 2Sx + V\right\}\right)$ where we define $V = (Y_1)^2 + (Y_2)^2 + \ldots + (Y_n)^2$. Let K $= \left(\frac{1}{\sqrt{2\pi}}\right)^n \exp\left(\frac{-1}{2}\left\{V - \frac{S^2}{n}\right\}\right)$ and notice that $\int_{\mathbf{R}} f_{N_1}(X+N_1-x)\cdot\cdot\cdot f_{N_n}(X+N_n-x)\, dF_X(x) =$

$K\int_{\mathbf{R}} \exp\left(-y^2/2\right) dF_X\left(\frac{y}{\sqrt{n}} + \frac{S}{n}\right)$; see for example [8, pp. 182-183]. Further, we have that

$\int_{\mathbf{R}} x\, f_{N_1}(X+N_1-x)\cdots f_{N_n}(X+N_n-x)\,dF_X(x) = \frac{K}{\sqrt{n}}\int_{\mathbf{R}} y\, \exp\left(-y^2/2\right) dF_X\left(\frac{y}{\sqrt{n}} + \frac{S}{n}\right) +$

$\frac{KS}{n}\int_{\mathbf{R}} \exp\left(-y^2/2\right) dF_X\left(\frac{y}{\sqrt{n}} + \frac{S}{n}\right)$. Using the above results, we see that

$$E[X|Y_1, \ldots, Y_n] = \frac{S}{n} + \frac{\int_{\mathbf{R}} z\, e^{-z^2/2}\, dF_X\left(\frac{z}{\sqrt{n}} + \frac{S}{n}\right)}{\sqrt{n}\int_{\mathbf{R}} e^{-z^2/2}\, dF_X\left(\frac{z}{\sqrt{n}} + \frac{S}{n}\right)} \text{ a. s.} \tag{1}$$

For the special case of $f_X(x) = \frac{1}{\sigma\sqrt{2\pi}}\exp(-x^2/2\sigma^2)$, we get that $E[X \mid Y_1, \ldots, Y_n] = S\sigma^2/(1 + \sigma^2 n)$ a.s. Furthermore, $E[X \mid Y_i] = (Y_i)\sigma^2/(1 + \sigma^2)$ a.s. Thus, we see that in this situation, $E[X \mid Y_1, \ldots, Y_n] = \left(\frac{1+\sigma^2}{1+n\sigma^2}\right)\sum_{i=1}^{n} E[X|Y_i]$ a.s.

Theorem 5: Consider a probability space $(\Omega, \mathcal{F}, P)$ and random variables $X, N_1, \ldots, N_n$ defined on $(\Omega, \mathcal{F}, P)$ where n is a positive integer and X is a second order random variable. Further, assume that $N_1, \ldots, N_n$ are identically distributed with zero mean, unit variance Gaussian distributions, and that $X, N_1, \ldots, N_n$ are mutually independent. Define $Y_i = X + N_i$ for $i = 1, \ldots, n$. Then there exists a Borel measurable function $g:\mathbf{R}^n \rightarrow \mathbf{R}$ such that $E[X|Y_1, \ldots, Y_n] = g(E[X|Y_1], \ldots, E[X|Y_n])$ a.s.

Proof: If X is a.s. equal to a constant, the result is obvious. Now, assume that X is not almost surely equal to a constant. Using (1) it immediately follows that the regression function $E[X|Y_i{=}y]$ is given by $\dfrac{\int_{\mathbf{R}} x \exp\left(\frac{-x^2}{2} + yx\right) dF_X(x)}{\int_{\mathbf{R}} \exp\left(\frac{-x^2}{2} + yx\right) dF_X(x)}$, a.e. Taking a derivative with respect to y, using Schwarz's inequality, and noting that X is not a.s. equal to a constant, one finds that $E[X|Y_i{=}y]$ is equal a.e. to a strictly increasing function of y. Hence, there exists a Borel measurable function g_i so that $g_i(E[X|Y_i]) = Y_i$ a.s. Note that

$S = \sum_{i=1}^{n} g_i(E[X|Y_i])$ a.s. Thus, substitution of the above expression for S into (1) results in a Borel measurable function of $g_1(E[X|Y_1]), g_2(E[X|Y_2]), \ldots, g_n(E[X|Y_n])$ which is equal to $E[X \mid Y_1, \ldots, Y_n]$ a.s.

Q.E.D.

Example 6: Let X have a uniform density on [−a, a], where a is a positive real number. It follows straightforwardly that

$$E[X \mid Y_1, \ldots, Y_n] = \frac{S}{n} + \frac{\left[\exp\left(-\frac{2}{n}(a + \frac{S}{n})^2\right) - \exp\left(-\frac{2}{n}(a - \frac{S}{n})^2\right)\right]}{\sqrt{\frac{n\pi}{2}}\left[\operatorname{erf}\left((a + \frac{S}{n})\sqrt{\frac{n}{2}}\right) + \operatorname{erf}\left((a - \frac{S}{n})\sqrt{\frac{n}{2}}\right)\right]} \text{ a.s.}$$

Further, we see that $E[X|Y_i] = Y_i + \dfrac{\left[\exp\left(-2(a + Y_i)^2\right) - \exp\left(-2(a - Y_i)^2\right)\right]}{\sqrt{\frac{\pi}{2}}\left[\operatorname{erf}\left(\frac{a + Y_i}{\sqrt{2}}\right) + \operatorname{erf}\left(\frac{a - Y_i}{\sqrt{2}}\right)\right]}$. Note that Theorem 5 implies that there exists a Borel measurable function g such that $E[X \mid Y_1, \ldots, Y_n] = g(E[X|Y_1], \ldots, E[X|Y_n])$ a.s.

CONCLUSION

In this paper we have considered the problem of fusion in estimation theory. We began by considering the best way to linearly combine the best minimum mean square linear estimates of a random variable X and the best way to linearly combine the best minimum mean square Borel measurable estimates of a random variable X based on a family of observations $\{Y_1, \ldots, Y_n\}$. We next presented several examples, using common distributions, in which virtually any method of fusion would be useless in approximating the random variable of interest. Finally, we presented a theorem which for a very general situation shows that fusion resulting in an almost surely exact approximation is always possible. In particular, this result addressed the situation in which the data consisted of the random variable of interest corrupted by additive Gaussian noise, and the random variable of interest could be any second order random variable.

ACKNOWLEDGEMENT

This research was supported by the Air Force Office of Scientific Research under Grant AFOSR-86-0026.

The authors also wish to acknowledge helpful comments on the subject of this paper from Prof. Bruce Hajek at the University of Illinois at Urbana-Champaign.

REFERENCES

[1] G.L. Wise, "A note on a common misconception in estimation," *Systems and Control Letters*, Vol. 5, pp. 355-356, April 1985.

[2] A. T. Alouani and J. D. Birdwell, "Distributed estimation: constraints on the choice of the local models," *IEEE Trans. Automatic Control,* vol. AC-33, pp. 503-506, May 1988.

[3] J. L. Speyer, "Computation and transmission requirements for a decentralized linear-quadratic-Gaussian control problem," *IEEE Trans. Automatic Control,* vol. AC-24, pp. 266-269, April 1979.

[4] D. A. Castanon and K. Teneketzis, "Distributed estimation algorithms for nonlinear systems," *IEEE Trans. Automatic Control,* vol. AC-20, pp. 418-425, May 1985.

[5] A. S. Willsky, M. G. Bello, D. A. Castanon, B. C. Levy, and G. C. Verghese, "Combining and updating of local estimates and regional maps along sets of one-dimensional tracks," *IEEE Trans. Automatic Control,* vol. AC-27, pp. 799-813, August 1982.

[6] D. A. Pierce and R. L. Dykstra, "Independence and the normal distribution," *The American Statistician,* vol. 23, no. 4, p. 39, 1969.

[7] W. Rudin, *Real and Complex Analysis,* third edition. McGraw-Hill: New York, 1988.

[8] N. Dunford and J. T. Schwartz, *Linear Operators Part I: General Theory.* Interscience Publishers: New York, 1957.

INDEX OF AUTHORS

Abed, E. H. 338
348
905
Agathoklis, P. 199
Allebach, J. 609
Anderson, B.D.O. 417
Antsaklis, P. J. 191
1160
1172
Apostolico, A. 253
Arbouz, N. M. 862
Arora, A. S. 1059
Atallah, M. J. 253
1114
Aupperle, L. J. 97
Azimi, M. 109
Baccelli, F. 965
Bai, S. 917
Ball, D. 169
Barmish, B. R. 187
Behroozi-Toosi, A. B. 979
Berger, T. 580
Bhargava, B. 368
Bhasker, J. 808
Bhattacharya, B. K. 107
Biglieri, E. 1080
Bilardi, G. 1084
Blake, I. 471
Blankenship, J. 378
Bovik, A. C. 239
Brahmadathan, K. 683
Brochin, F. M. 1001
Bross, A. 119
Browne, S. 368
Bucklew, J. A. 556
Bui, T. N. 798
Burnashev, M. V. 1059
Butler, J. T. 408
Calderbank, A. R. 1082
Callahan, K. R. 758
Cassandras, C. G. 1170
Catarci, T. 1138
Chakrabarti, C. 1015
Chakravarty, S. 740
Chandran, S. 87
Chandrasekharan, N. 283
Chang, B. 459
Chao, C. C. 1049
Chao, P. C. 1037
Chen, J. 881
Chen, L. 264
Cheng, W.-T. 750
Chern, M.-Y. 1037
Chiang, Y. W. 969
Choe, Y. 647
Chung, H. 494
Coan, B. A. 663
Cohn, D. L. 1059
Conn, H. E. 97
Costello, Jr., D. J. 1061
Coyle, E. J. 589
Crescenzi, P. 721
Dandapani, R. 758
Das, M. 629
Dasgupta, S. 229
Dattatreya, G. R. 15
Davison, E. J. 907
DeCarlo, R. 1150
Dehne, F. 1035
Delosme, J.-M. 1104
Delp, E. J. 657
Douligeris, C. 1183
Drakopoulos, E. 39
Dubois, M. 701
Erickson, C. 109
Esfandiari, F. 535
Farison, J. B. 893
Farrell, J.A. 852
Feig, E. 473
Ferreira, H. C. 1061
Forsberg, K. 471
Frazho, A. E. 872
Freudenberg, J. S. 881
Fu, J.-H. 338
Fuhrmann, D. R. 219
Fuja, T. 461
Gambosi, G. 819
Gao, Z. 191
Garber, F. D. 11
Gawronski, W. 209
Georghiades, C. N. 1
Geyer-Schulz, A. 810
Goel, D. K. 758
Golic, J. Dj. 639
Gopal, G. 991
Greenberg, A. G. 571
967
Gubner, J. A. 499

Gulati, R. K. 758
Guo, T. 356
Haddad, A. H. 937
Hall, E. B. 599
Hambrusch, S. E. 1114
Hedetniemi, S. T. 283
Heegard, C. 461
Herley, K. T. 1084
Herro, M. A. 1082
Hsu, W.-L. 842
Hu, Y. H. 1042
1046
Huang, Y.-F. 229
Hui, S. 518
Hunt, III, H. B. 788
Hwang, G.-Y. 1126
Iftar, A. 883
Ingram, M. A. 937
Italiano, G. F. 819
JaJa, J. 1015
Janko, W. H. 810
Jha, N. K. 768
Johnson, C. D. 159
Johnson, C. R. 417
Johnson, R. 274
Jou, J.-Y. 750
Juang, J.-N. 209
Jung, D.-M. 83
Kamen, E. W. 437
Kanal, L. N. 15
Kanefsky, M. 459
Kappos, E. 953
Kashyap, R.L. 647
Katsaggelos, A.K. 661
Keil, J. M. 97
Kermani, P. 566
Kesler, S. B. 457
Khalil, H.K. 535
Kinney, L. L. 778
Kolla, S. R. 893
Koppelman, D. M. 981
Krenz, G. S. 313
Kudva, P. 547
Kumar, D. 139
Kumar, P. V. 492
494
Lai, T.-H. 1094
Lakshman, T. V. 673
Larmore, L. L. 253
Lay, K.T. 661
Le, D. K. 872
Lee, C. C. 39
969
Lee, D. T. 129
Lee, P.-N. 378
Lee, S. C. 1005
Lei, T. 659
Li, J.-H. 508
Li, Y. 1140
Lin, Q. 609
Liu, C.-M. 492
Liu, L. 119
Liu, R.-W. 331
731
Liu, Z. 965
Logan, D. 119
Loh, N. K. 629
Longman, R. W. 927
Lou, R. D. 1136
Lu, H. 1005
Madani-Esfahani, S. M. 518
Madisetti, V. K. 149
Malladi, S. S. 1061
Manacher, G. K. 832
Mankus, T. A. 832
Mareels, I. M. Y. 417
Marinescu, D. C. 955
Massey, W. A. 967
Mazumdar, R. 1183
McClamroch, N. H. 356
537
McEliece, R. J. 1049
McFaddin, H. S. 253
McLane, P. J. 1080
Melhem, R. 1126
Messerschmitt, D. G. 149
Metzner, J. J. 482
Michel, A. N. 508
852
Milenkovic, P. H. 1042
Miller, D. E. 907
Miller, M. I. 1040
Miller, R. K. 313
Milosavljevic, M. M. 251
Moitra, A. 274
Moitra, D. 303
Moulin, P. 27
Mount, D. M. 87
Murthy, G. R. 589
Naganathan, S. 1046
Nakajima, K. 398
Nandy, S. K. 1025
Narasimhamurthi, N. 356
Nayeri, M. 447
Ney, P. 556
O'Rourke, J. 49

97
O'Sullivan, J. A. 27
Oloomi, H. 903
Oruc, A. Y. 981
Ozguner, U. 883
Panwar, R. B. 1025
1116
Passino, K. M. 1160
1172
Patnaik, L. M. 1116
Paulik, M. J. 629
Pawlak, M. 366
Peck, A. 798
Peczkowski, J. L. 177
Peir, J.-K. 119
Peleties, P. 1150
Pham, Q. T. 1035
Phan, M. 927
Porod, W. 508
Prince, J. L. 619
Ramakrishnan, T. 778
Raman, V. 331
Ramanan, P. 58
Rao, A. K. 229
Rao, R. R. 979
Ravi, S. S. 388
788
Rodger, S. H. 293
Rosberg, Z. 566
Rosen, Y. 408
Rosenkrantz, D. J. 388
Roysam, B. 1040
Sadowsky, J. S. 556
Sain, M. K. 177
189
Salam, F. M. A. 917
Sallam, B. M. 457
Salowe, J. S. 73
Samad, T. 808
Sanchis, L. A. 711
Sandberg, I. W. 169
Sanders, K. D. 239
Sarrafzadeh, M. 1136
Sartori, M. A. 1172
Sastry, C. R. 437
Sawan, M. E. 903
Saydy, L. 348
Schevon, C. 49
Schrader, C. B. 189
Schulz, T. J. 13
Sennott, L. I. 569
Sethares, W.A. 417
Sewchand, W. 659
Shen, D. 119
Sherman, D. L. 249
Shi, Z. 187
Shih, F.-H. W. 398
Silvestri, R. 721
Simaan, M. 437
939
Sira-Ramirez, H. 321
Skavantzos, A. 1044
Snorrason, O. 11
Snyder, D. L. 13
27
Sreeram, V. 199
Srivastava, P. 25
Stark, W. E. 1070
Stearns, R. E. 788
Stojmenovic, I. 1035
Stouraitis, T. 1044
Strickland, S. G. 1170
Su, R. 862
Sugihara, K. 63
Suzuki, I. 63
Syrmos, G. L. 905
Taipale, D. 480
Talamo, M. 819
Tamboli, A. 378
Tan, H. L. 657
Taudes, A. 810
Telang, V. 1082
TeWinkel, L. E. 1114
Thomas, J. B. 1001
Tits, A. L. 348
Tong, L. 731
Tsai, K. 842
Uresin, A. 701
Vanderbei, R. J. 571
Vastola, K. S. 37
Venkatesan, S. 683
Vidyasankar, K. 693
Wang, C. A. 85
Wang, D. 537
Warren, D. J. 35
Wei, V. K. 673
Weinrib, A. 991
Wessel, A. E. 599
White, W. 1094
Willett, P. 25
Willsky, A. S. 619
Wise, G. L. 599
Woerner, B. D. 1070
Wolf, J. K. 1048
Wong, Y. 1104
Wonham, W. M. 1140

Wu, Q. 219
Wyman, B. F. 177
Yalamanchili, S. 129
Yamashita, M. 63
Yang, J. C. S. 547
Yang, T.-C. 547
Yanushevsky, R. T. 501
Yaz, E. 948
Ye, Z. 580
Yesha, Y. 264
Yin, G. 427
Young, K. D. 528
Yousef, H. 939
Zak, S. H. 518
Zazanis, M. A. 549
Zhang, L. 731
Zhu, J. 159
Zoltowski, M. D. 249

UNIVERSITY OF ILLINOIS-URBANA
621.3811AL5 C001
PROCEEDINGS / ALLERTON CONFERENCE ON COM
1988:1
3 0112 008149244